Ole Henrik Aarum Kolstad

Model for Detecting Flaws in Railway Rails using Machine Learning

Master's thesis in Engineering and ICT
Supervisor: Jørn Vatn
June 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Kunnskap for en bedre verden

Ole Henrik Aarum Kolstad

Model for Detecting Flaws in Railway Rails using Machine Learning

Master's thesis in Engineering and ICT
Supervisor: Jørn Vatn
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Preface

This master's thesis is submitted to the Norwegian University of Science and Technology as the concluding part of the Master of Science in Engineering and ICT at the Department of Mechanical and Industrial Engineering. The study was conducted during the spring semester of 2019. Figures in this thesis contain colours crucial to providing the information intended. As a result, reading this thesis in black and white colours only is strongly discouraged.

I would like to express sincere gratitude to my supervisor professor Jørn Vatn for invaluable feedback and helpful discussions. This thesis would not be possible without his efforts and industry network. Furthermore, I would like to thank Head of Non-Destructive Testing at Bane NOR, Harald Schjelderup, for his willingness to provide data and information, in addition to making himself available for questions through multiple interviews. Finally, I want to thank my family, friends and classmates for continued support and guidance throughout the student years.

Trondheim, 2019-06

Ole H. Kolstad

Ole Henrik Aarum Kolstad

Summary

Railway companies responsible for railway infrastructure hire inspection companies to inspect the rails for suspected flaws. The inspection companies use trains to scan the rails using ultrasound at relatively high speeds. Subsequently, the railway companies conduct manual inspections of the suspected flaws, due to the initial inspection not being sufficiently reliable. The number of false positives, i.e. suspected flaws not found during manual inspection, among the suspected flaws is high and are cause for unnecessary manual inspections. The railway companies receive predicted properties and visual representations of ultrasound signals, called "B-scans", of the suspected flaws. The aim of the thesis was to lower the amount of false positives by developing models, using machine learning, able to detect whether the ultrasound signals in a B-scan represent a true flaw.

Case company Bane NOR provided inspection data and B-scans. Analysis of the provided data and semi-structured interviews with a case company contact was conducted to establish the details of the case. The supervised learning method of classification was used to train the models to recognize patterns in the B-scans indicative of flaws. Before subjected to training, a subset of B-scans was selected from the complete dataset and pre-processed. 7 steps of pre-processing was completed to transform the B-scans provided into a format suitable for machine learning; extracting the correct B-scan, removing grid lines, extracting sensor subsets, removing colour, reducing resolution, feature selection and binary conversion. The models were trained using the Random forest algorithm and the parameter of number of trees was tuned. 21 different combinations of subsets and parameter values were trained 3 times each to evaluate the average performance. 81 models were trained in total and the performance was evaluated using confusion matrix performance measures and Precision, Recall and F1.

Models were successfully developed to detect flaws in railway rails and the advantages and disadvantages of the models are discussed. The optimal model depend on the situation and three scenarios are presented. The top performing model achieved a 93.1% accuracy of detecting flaws with a 36.9% reduction in false positives, i.e. redundant inspections saved. Performance was found to improve with increased number of samples and decreased number of features. Further work is suggested and requested to improve the results and apply the models.

Contents

Chapter 1

Introduction

1.1 Background

Maintenance of railway infrastructure is characterized by large budgets and frequent maintenance (Zerbst et al., 2009a). Additionally, conflicting demands for continuing operations and conducting maintenance is prevalent (Lidén, 2015). Failure to maintain the railways can result in injuries and loss of life, and a study on Swedish railways found that 30% of railway incidents were related to maintenance (BBC, 2017; Holmgren, 2005). The rails are particularly vulnerable due to high service loads and harsh environmental conditions (Zerbst et al., 2009b).

To determine the condition of rails, companies specialized in rail inspection are hired by railway companies responsible for the infrastructure. The inspection companies use high-speed test trains operating between 40 km/h and 100 km/h to scan the rails using ultrasound (Papaelias et al., 2008). After the initial inspection, the inspection companies provide the railway companies with the locations and properties of suspected flaws found during the inspection. The railway companies also receive two-dimensional visualizations of the ultrasound signals indicating flaws, called "B-scans" (Clark, 2004). The railway companies find the process to be insufficiently reliable by itself. As a result, manual inspection of the suspected flaws are conducted by the railway companies. A significant amount of suspected flaws are not found during manual inspection. Due to such "false positives", redundant manual inspections are conducted. Furthermore, false positives often outnumber true positives on the rail network (Papaelias et al., 2008). The number of manual inspections needed could be lowered by developing an accurate model for flaw detection.

Machine learning has proven effective for recognizing complex patterns in data and is particularly suitably when the system is too complex to manually design (Bishop, 2006; M Mitchell, 2006). The machine learning problem is described in Suthaharan (2016a) as how to fit a model between a data set and its corresponding response set, and how to train and validate the model to learn the system's characteristics from data. Models have previously been developed using machine learning for comparable problems. B-scans are created from "A-scans", a more raw data form displaying the amplitude of the ultrasound signals (Cygan et al., 2003). Machine learning models have previously been trained on A-scans to classify flaws in rail welds (Singh and Manning, 1983; Chen et al., 2014), estimation of crack depth and inclination (Takadoya et al., 1995; Kitahara et al., 1992), and crack size and orientation (Zgonc et al., 1995). Additionally, machine learning models have been developed to detect surface defects in rails from image data (Hajizadeh et al., 2016). A hybrid system based on rules and case-based reasoning has previously been proposed for detecting flaws from B-scans (Jarmulak et al., 1998). The system was able to detect 27% of the flaws, while the remaining samples were flagged as needing further inspection. As a result, the system would not lower the amount of false positives. To the best of the authors knowledge, no other literature cover rail flaw detection using machine learning on B-scans. There is a lack of literature acknowledging and proposing solutions to the problem experienced by railway companies. Research is needed on detecting rail flaws from B-scans, as the railway companies do not have access to A-scans. This thesis cover the process of developing machine learning models for detecting rail flaws from B-scans.

1.2 Problem Formulation

The aim of the thesis was to explore the potential of using machine learning to develop models capable of detecting rail flaws from B-scans. Information and data for the problem was provided by case company Bane NOR. An initial interview was conducted with a Bane NOR contact to specify the problem, their current solution and the potential of an improved solution. It was established that Bane NOR experts are not able to accurately identify flaws from the B-scans provided by the inspection companies. As a result, Bane NOR sends a team for conducting manual inspections. The team consist of 2-3 employees dedicated to manual inspections full time from April until October. During their manual inspection, suspected flaws are registered as either found, not found or below the threshold for registration. An initial analysis of the inspection data provided by Bane NOR was conducted to evaluate the problem. 18.4% of the samples were false positives if regarding those below the threshold as flaws. If the flaws below the threshold were included as false positives, the amount became 39.3%. Considerable time and resources could thus be saved by reducing the rate of false positives.

Machine learning was chosen to develop the model due to the ability of learning from data, having satisfying amounts of data available and being used successfully for A-scans. Bane NOR does not currently employ machine learning techniques in any projects known to the Bane NOR contact. During preliminary review, the B-scans provided by the case company were found to be in need of pre-processing to extract the relevant data.

1.3 Objectives

The objectives to be completed to achieve the aim of the thesis were:

- Establish case specifics through interviews
- Identify data characteristics through data analysis
- Select a data subset
- Pre-process the B-scans
- Choose the appropriate algorithm and performance measures
- Train and evaluate machine learning models

The main research question to be discussed after completing the objectives is:

1. What are the advantages and disadvantages of the proposed model(s)?

1.4 Methodology

A case study was conducted to evaluate a machine learning approach to the flaw detection problem, by testing on empirical ultrasound data provided by case company Bane NOR. The data was originally collected by rail inspection companies Sperry Rail Services and EURAILSCOUT, in Norway between 2003 and 2019. Inspection data for every suspected flaw was provided by Bane NOR as an Excel sheet. The development of machine learning models require several decisions to be made, e.g. choosing the appropriate algorithm and performance measures (Du and Swamy, 2014). The optimal choices are considered to be problem-specific and therefore the details of the case had to be explored and described. Information was gathered through both qualitative and quantitative methods to perform informed decisions: 2 semi-structured interviews and data analysis. A case company contact was interviewed to gain expert knowledge on the data and problem. The data was analyzed using code written in Python and Microsoft Excel to tailor the approach to the case. Additionally, results in literature for comparable problems were searched for using Google Scholar. The method applied for developing the models is detailed in the case study chapter.

1.5 Limitations

Ultrasound signal data is commonly represented in three different ways: A-scan, B-scan and C-scan (Cygan et al., 2003; Gordon et al., 1993). The scope was limited to B-scans and prediction of the flaw status of the rail, not other properties such as flaw size, orientation or severity. The time available for the study was 140 days or 20 weeks. The process of developing the model was explored, but not the process of implementation. Furthermore, machine learning is a broad term with a wide range of applications and algorithms. A single machine learning algorithm was used to develop the models. The thesis does not include a review and comparison of different algorithms, which is covered in literature (Lim et al., 2000). The algorithm used was Random Forest (Breiman, 2001). Part of the provided data originally included names of inspection operators. The names have been anonymized to protect individual privacy.

1.6 Structure

The thesis is structured as follows:

- **Preface:** Contains practical information and acknowledgements.
- **Summary:** Short version of the thesis with key takeaways presented.
- **Introduction:** Background for thesis and problem presented. The research methodology used and limitations of the study.
- **Theory:** Provides the theoretical background within rail inspection and machine learning needed to understand the following chapters.
- **Case Study:** Describes the process used for pre-processing the B-scans and developing the models.
- **Results:** Model performances are presented and compared.
- **Discussion:** Results and findings are discussed and methods used are evaluated.
- **Conclusion:** Main results are summarized and improvements and further work requested.
- **Bibliography**
- **Appendix A:** Acronyms
- **Appendix B:** Inspection Data
- **Appendix C:** Individual Model Performance

Chapter 2

Theory

This chapter introduce the terminology and basic knowledge within rail inspection and machine learning needed to understand the following chapters. Due to extensive numbers of methods within both topics, this chapter aim to provide theory relevant to the specific case in the study. The case specifics are explored and described in the case study chapter, along with reasoning for decisions made. As a result, the topic of rail inspection is concentrated on rail flaws and the ultrasound inspection method. The topic of machine learning primarily covers methods specific to supervised learning and classification.

2.1 Rail Inspection

The elements of railway tracks may differ in scale and material, but is largely consistent. Figure 2.1 describes the components of a railway track. Crossties, also called sleepers, have historically been built out of timber, but steel reinforced concrete has become the preferred material during the last half century (Giben et al., 2015). The crossties distribute wheel loads from the rail feet to the ballast (Kaewunruen et al., 2016). The ballast is placed between the crossties and the soil below and consist of granular materials, e.g. rock masses. The ballast distributes the load impact from the crossties across the low strength soil underneath (Tutumluer et al., 2006). The fasteners ensure good track geometry crucial for stability by fastening the rails to the crosstie (Hong et al., 2018). The rails guide the train cars and are the elements acting between the train wheels and the railway. The rails are thus subject to contact fatigue and wear (Wang et al., 2003). The material currently used for producing rails are different types of steel (Baptista et al., 2018). Cast iron was previously used, but proved susceptible to rail failures. The cast iron is a brittle material not able to sufficiently redistribute the loads (CANNON et al., 2003).

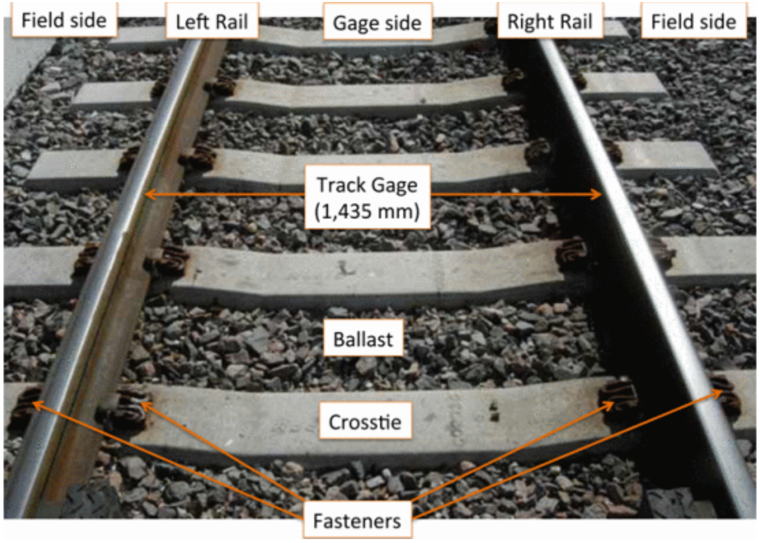


Figure 2.1: Definition of railway track elements (Giben et al., 2015)

The rails absorb all vertical and lateral forces and can be viewed as the vital part of the railway track (Faiz and Singh, 2009). As a result, rails are in need of frequent inspections. Cost of rail failures include rail inspections, train delays, replacement and repairs, preventive measures, derailments and loss of public confidence (CANNON et al., 2003). Figure 2.2 illustrate the different parts of a rail in profile. The head surface is in contact with the train wheels and therefore subject to surface flaws. The outer sides of the head are called gauges. The head is connected to the foot by the web. The foot is fastened to the crosstie as illustrated in Figure 2.1.

Forces and stresses applied to rails have a directional element. Additionally, flaws in rails, especially cracks, occur and propagate in different directions. For simplicity, the directions vertical, longitudinal and lateral are visualized in Figure 2.2. The direction for travel of rail vehicles and the extension of the rails are in the longitudinal direction. The vertical axis cover the rail element from the foot to the head surface. Lastly, the lateral direction follow the width of the rail.

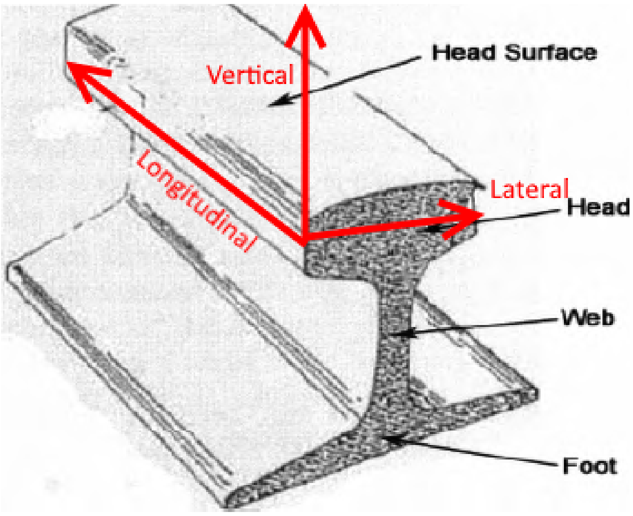


Figure 2.2: Rail parts and directions illustrated (Lin Jie et al., 2009)

To join two rails, boltholes have traditionally been machined into the rail web and joined using a fishplate, see Figure 2.3. The boltholes are susceptible to flaws, and the fishplate method has been largely replaced by continuous welding to reduce the number of boltholes, see Figure 2.4 (CANNON et al., 2003; Maruyama et al., 2008).

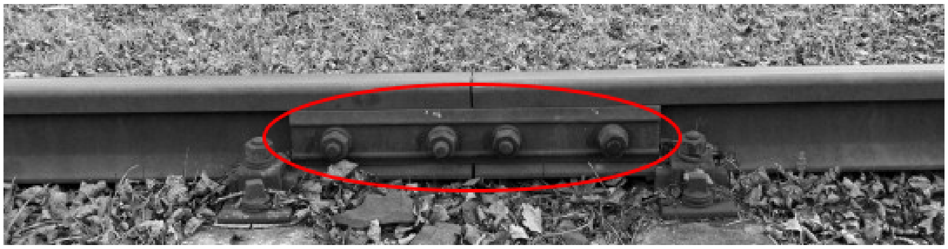


Figure 2.3: Fishplate joint (Zerbst et al., 2009a)

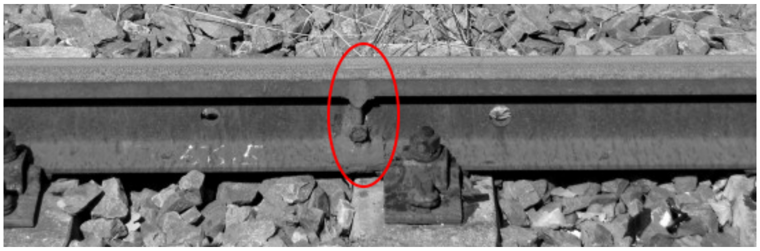


Figure 2.4: Welded joint (Zerbst et al., 2009a)

2.1.1 Forces Applied to Rails

A complex combination of loads result in stress in rails. Particularly cyclic stresses including thermal stresses, bending stresses and contact stresses often dominate rail stresses and lead to crack growth and fatigue failure (Greisen et al., 2009). The different types or stresses are illustrated in Figure 2.5. Forces applied to rails have been categorized into bending stresses, shear stresses, contact stresses, thermal stresses, residual stresses and dynamic effects (Zerbst et al., 2009a; CANNON et al., 2003). The following paragraphs in this subsection include short introductions of the different stresses and effects.

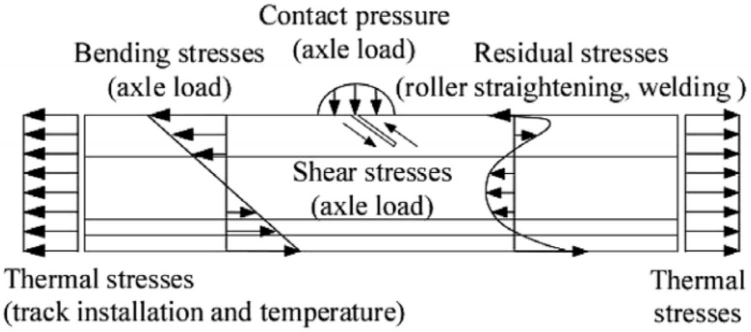


Figure 2.5: Forces applied to rails (Zerbst et al., 2009a)

Bending and Shear Stresses Rail bending due to wheel load is comprised of a vertical and a lateral component (Zerbst et al., 2009a). While the lateral component contribute to rail failure, the vertical component dominate. The stresses compress in the rail head and tensile in the rail base. The design of the rail profile minimize the effect on the rail base from rail head wear (CANNON et al., 2003). The wheel loads also generate shear stresses in the rail (Zerbst et al., 2009a).

Contact stresses The contact between wheel and rail result in contact stresses (Zerbst et al., 2009a). The stresses originate from forces due to wheel load and traction, including braking and steering. Poorly conforming wheels and rails can significantly increase the already high contact stresses (CANNON et al., 2003).

Thermal Stresses High temperatures result in thermal stresses and are typically caused by sun exposure or high speeds. Similarly, thermal stresses caused by low temperatures deal added stress to the rail (Zerbst et al., 2009a).

Residual Stresses Stresses occurring in rails not subjected to external load are residual stresses. These types of stresses originate from the manufacturing of the rails and are caused by heat treatment, roller straightening and welding (Zerbst et al., 2009a). Residual stresses are generally tensile and prominent in the centre of the rail head, contributing to flaw initiation and growth (CANNON et al., 2003).

Dynamic Effects The motions of train cars and the dynamic response of the track cause dynamic effects (Zerbst et al., 2009a). Irregularities in rails and wheels are generally the origin, and are typically caused by wheels that stop rotating due to braking (CANNON et al., 2003).

2.1.2 Types of Flaws

The forces applied to rails can result in different types of flaws in the rails. The following paragraphs in this subsection include short introductions of the different types of flaws.

Surface Flaws Rolling contact fatigue (RCF) are the most common flaws and caused by contact stresses between the wheel and rail, causing severe shearing of the surface layer and fatigue or exhaustion of the steel. RCF is predicted to be a growing concern in the future, as demand for higher speeds, axel loads, traffic density and traction increase (CANNON et al., 2003). Damage to the surface layers can also be caused by errors in installation and use. An example is "spalling", see Figure 2.6, caused by wheels spinning.

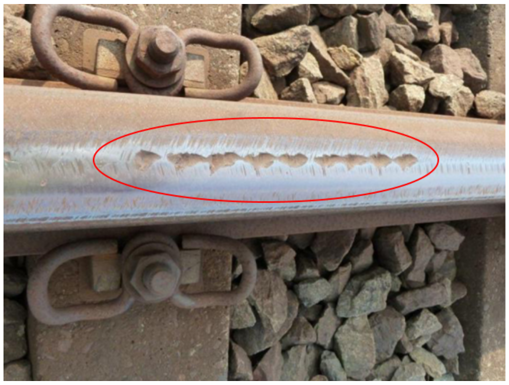


Figure 2.6: Spalling surface flaw (Steenbergen, 2016)

Cracks Originating from Surface Cracks are often initiated by surface flaws in the rail head surface and propagate through the surface layers. The type of cracks include "head checks" and "squats". Such cracks can hide deeper and potentially more dangerous cracks if using ultrasonic inspection methods (CANNON et al., 2003). Head checks and squats often occur in multiples, which makes them particularly dangerous as crack propagation could lead to damages over an extended distance. Risk of derailment is greatly increased if parts of or all of the rail head surface lacks, and especially over longer distances (Zerbst et al., 2009a). Figure 2.7 depicts a head check.

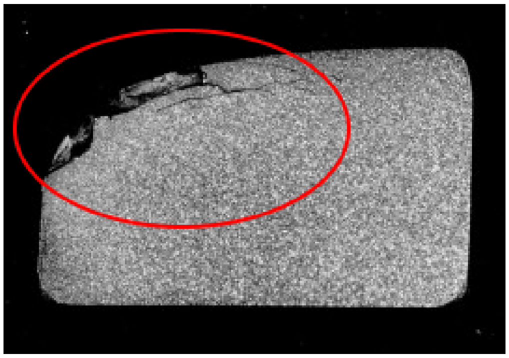


Figure 2.7: Transverse view of head check (Zerbst et al., 2009a)

Squats occur in straight or slightly curved tracks, dissimilar to head checks, and at the rail head surface, not the gauges. Unlike head checks, they occur randomly at isolated locations (CANNON et al., 2003). Cracks caused by head checks propagate in the lateral direction first, while those cause by squats propagate in the longitudinal direction first. They then both propagate transversely (Zerbst et al., 2009a).

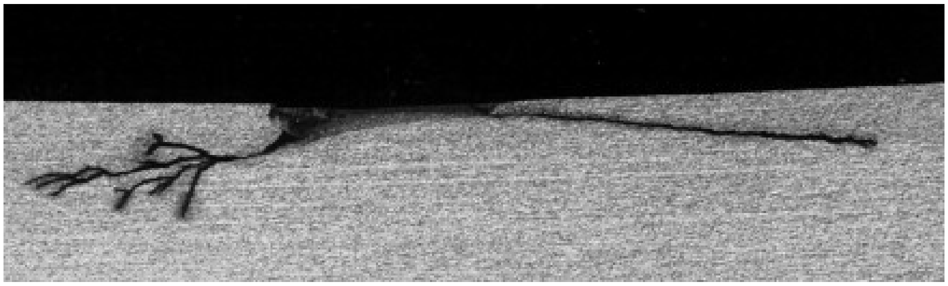


Figure 2.8: Squat crack propagating in the longitudinal direction (Zerbst et al., 2009a)

Despite surface flaws and surface-initiated cracks typically being a result of errors in installation or use, including traffic frequency and wheel loads, efforts can be made by railway infrastructure companies and railway service companies to resolve them (CANNON et al., 2003). Surface flaws can often be contained by grinding the rail head surface to remove the damaged areas or use a different type of rail head transverse profile to reduce contact stress or improve steering (Steenbergen, 2016). Figure 2.9 display a rail subjected to grinding. Surface flaws can also be limited by reducing the traction between wheel and rail in curves using lubrication. Water entrapment has probably the most significant influence on shallow crack growth initiated by surface flaws (CANNON et al., 2003). Water entrapment is caused by water leaking into the crack, causing pressure exerted at the crack tip when forces are applied from wheel load (Macha et al., 1999).

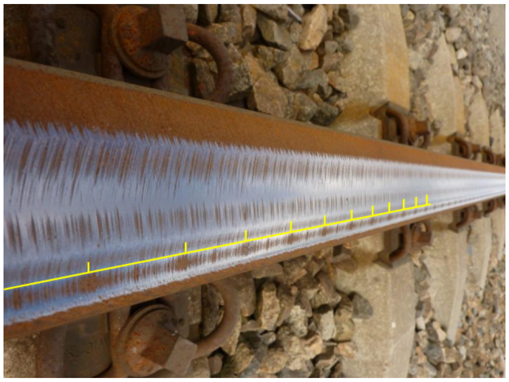


Figure 2.9: Rail after grinding operation (Steenbergen, 2016)

Cracks Originating Internally Kidney cracks, see Figure 2.10, are cracks that do not originate from surface flaws, but internally in the rail (Zerbst et al., 2009a). The cracks are typically caused by manufacturing flaws, such as hydrogen shatter cracks caused by too high levels of hydrogen in the steel-making process (CANNON et al., 2003; Dayal and Parvathavarthini, 2003).

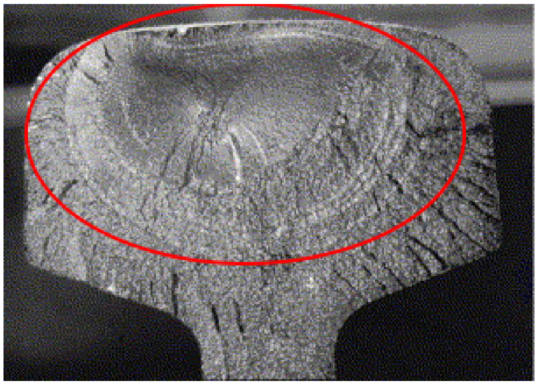


Figure 2.10: Kidney crack in traverse section (Steenbergen, 2016)

Longitudinal cracks in the rail head are called vertical split heads (VSH). These cracks spread vertically near the middle of the head (Toliyat et al., 2003). Longitudinal cracks in the gauges, see Figure 2.11, can result in material breaking off, such as gauge corner shelling and transverse cracks. Internal cracks are generally caused by manufacturing flaws (Zerbst et al., 2009a). Cracks can also form internally in the rail web, both longitudinal, lateral and vertically. Figure 2.12 depicts a crack that has propagated in the longitudinal direction of the web.

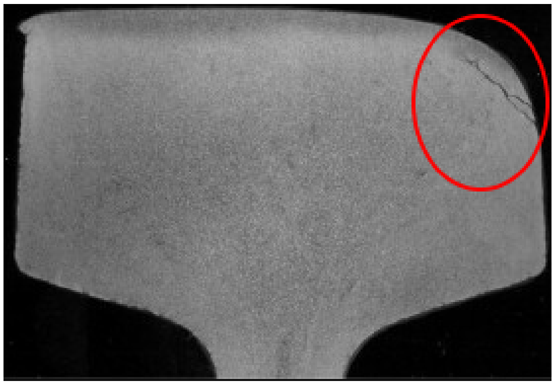


Figure 2.11: Vertical split head in gauge (Zerbst et al., 2009a)

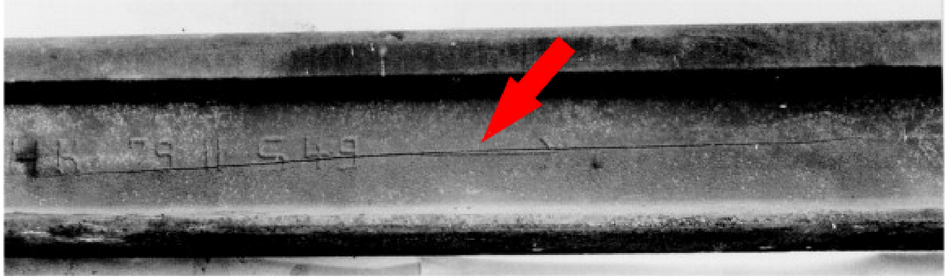


Figure 2.12: Longitudinal crack in web (Zerbst et al., 2009a)

Another type of cracks originating internally are bolt hole cracks, see Figure 2.13. Shear stresses are the main cause of failures at the boltholes in the rail web and often originate from fishplate restraints (Orringer et al., 1984; Zerbst et al., 2009a). The cracks are considered critical as they typically occur near the rail ends, weakening the part significantly (Zerbst et al., 2009a).

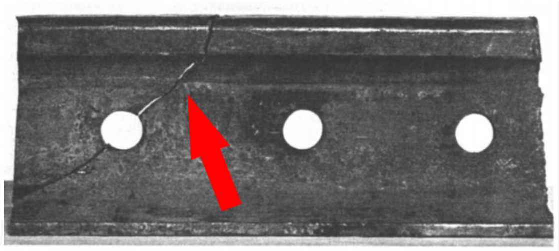


Figure 2.13: Bolt hole crack (Orringer et al., 1984)

Rail foot cracks can propagate both transversal and longitudinal. Transverse cracks are caused by wear or corrosion at the rail support, while longitudinal cracks are caused by manufacturing flaws (Zerbst et al., 2009a). A transverse crack originating from the foot underside is shown in Figure 2.14. Cracks caused by manufacturing flaws are addressed by the manufacturers through improvements in steel and rail-making technology. Rail flaws that occur in clusters over a distance between a few millimetres and several meters, such as longitudinal cracks, are critical flaws. Isolated transverse fractures are less likely to cause derailment (CANNON et al., 2003).

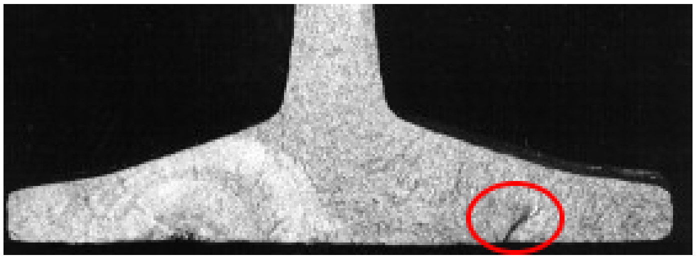


Figure 2.14: Traverse section of transverse crack (Zerbst et al., 2009a)

Areas of the rail containing switches or welds are more prone to flaws. Modification of the material properties, added residual stresses and lack of straightness and alignment of the rails can be caused by welding. The surface of the rails can be strongly deformed and lead to additional bending stresses in the foot, making cracks grow from underneath (Zerbst et al., 2009a).

Flaws caused by Extreme Temperature High temperatures can result in expansion of the rail material, with risk of buckling, i.e. straight rails becoming curved. To combat this problem, rails are welded under conditions that simulate high ambient temperatures. As a result, the rails are in tension for most of the year (CANNON et al., 2003). The disadvantage of the solution is increased risk of rail fracture during cold winter nights (Zerbst et al., 2009a).

2.1.3 Inspection Techniques

Non-Destructive Evaluation (NDE) inspections are conducted without invasive procedures. Various NDE techniques are employed for inspecting internal and external flaws to rails. These include ultrasonic probes, magnetic induction, pulsed eddy current (PEC), visual cameras and radiography. Inspections using ultrasonic has been dominating for high speed NDE since its introduction in 1953 (Papaelias et al., 2008).

During ultrasound inspection, a beam of ultrasonic waves is transmitted into the rail by ultrasonic probes (Clark, 2004). Ultrasonic waves are acoustic waves with a frequency above 20 000 Hz (Krautkrämer and Krautkrämer, 2013). The waves are generated by probes able to both emit and receive (Cygan et al., 2003). The ultrasonic waves are coupled into the rails by using a liquid, typically water, and the rail surface is optimally completely flat (Gordon et al., 1993). Special sprinklers are used to spray the liquid onto the rail (Papaelias et al., 2008). Information of irregularities in the material are obtained by analyzing the reflected wave, called echo, or by its shadow (Krautkrämer and Krautkrämer, 2013). A reflective wave occurs if the wave flow is disrupted by a change in geometry, e.g. the presence of a crack, and thus reflected back to the probe (Cygan et al., 2003). Total shadow zones can occur behind a crack because of the wave reflected as on a mirror (Krautkrämer and Krautkrämer, 2013). The location, type of flaw and structural integrity of the rail can be distinguished by analysis of the amplitude of the reflections and at what time they occur (Clark, 2004). To cover the different parts of the rail profile, the probes are placed in various angles. Common angles include 0°, 37° or 45° and 70°, see Figure 2.15. Ultrasonic probes can also be positioned to look across the rail head for squats and VSH (Clark, 2004).

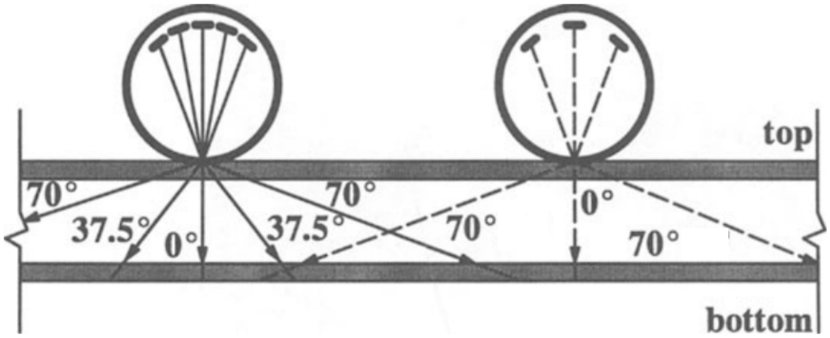


Figure 2.15: Probe arrangement in different angles (Chou et al., 1999)

Ultrasonic inspections can be performed using high-speed test trains equipped with ultrasonic probes or manually with portable ultrasound equipment. Figure 2.16 display a portable ultrasound device. Inspection speeds vary between 40 km/h and 100 km/h for test trains, while manual inspections with portable devices are conducted in walking speeds (Papaelias et al., 2008).

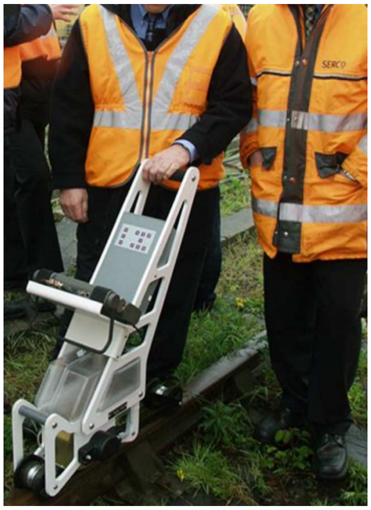


Figure 2.16: Portable ultrasound device (Clark, 2004)

The performance of detecting deep surface-breaking and internal flaws in the rail head and web are generally good when conducting high-speed inspections (Papaelias et al., 2008). Surface flaws smaller than 4 millimetre are typically not detected and small surface flaws can shadow larger and more critical internal flaws (Krautkrämer and Krautkrämer, 2013). Better ultrasonic probe arrangements and use of PEC are investigated as possible solutions (CANNON et al., 2003). Flaws in the rail foot are often missed as the foot is only partially scanned (Papaelias et al., 2008). Another problem is the poor performance when inspecting welds because the welds interfere with the ultrasonic waves (Clark and Singh, 2003).

2.1.4 Ultrasound Signal Representations

The ultrasound signals can be presented in various ways. The one-dimensional A-scan are visualizing the ultrasonic echos from reflective waves (Cygan et al., 2003). The horizontal axis represent the propagation time of the wave, while the vertical axis represent the amplitude of the reflective wave. Figure 2.17 depicts an A-scan where the first spike indicate the surface, while the second and third indicate a notch in the steel (Hall, 1976). The horizontal axis thus correspond to the vertical range of the rail and the distance between the second and third spike indicate the vertical size of the crack. If the amplitude reach a set threshold, a crack is anticipated. The threshold needs to be adjusted to avoid false positives, i.e. false indications of flaws, but still be able to detect actual cracks (Cygan et al., 2003). A time window is typically set to remove the initial amplitude caused by the surface (Clark, 2004).

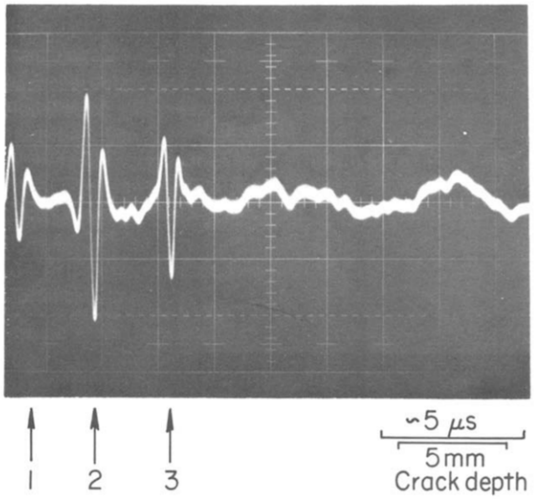


Figure 2.17: Example of A-scan with numbered spikes in amplitude (Hall, 1976)

Two-dimensional B-scans are constructed by combining A-scans (Cygan et al., 2003). The B-scans are easier to interpret, because they visualize the geometry of the track (Clark, 2004). The amplitude of the A-scans are in binary representation, either above the threshold or not (Cygan et al., 2003). The horizontal axis represent the longitudinal length of the rail, while the vertical axis represent the depth of the rail (Cygan et al., 2003; Clark, 2004). Figure 2.18 visualize how A-scans are combined into a B-scan. The A-scans seen in Figure 2.18 are similar to the A-scan in Figure 2.17, but rotated 90° clockwise. Filtering algorithms are typically applied to further increase readability by removing unnecessary data points (Clark, 2004). Similar to the B-scans presenting the cracks from a side view of the rail, a C-scan is a top-down view on the rail head surface. The C-scans are, similarly to B-scans, constructed from A-scans (Gordon et al., 1993).

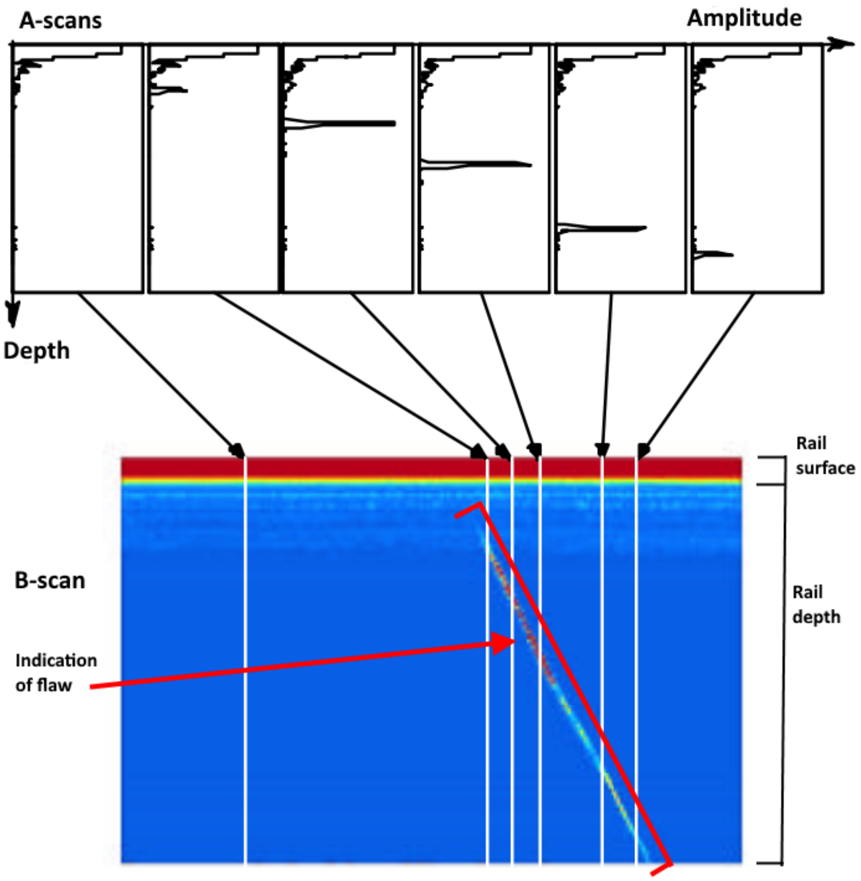


Figure 2.18: Process of converting several A-scans into a B-scan (Cygan et al., 2003)

2.2 Machine Learning

In the context of computer science, algorithms are used for solving problems on a computer. An algorithm is a sequence of actions to carry out in order to transform the input to output. Algorithms are easily defined for basic tasks such as sorting numbers in ascending order. However, the appropriate algorithm is not necessarily known for the problem at hand. Some problems are deemed too complex or labour intensive for constructing accurate algorithms (Alpaydin, 2014). Lack of knowledge and vast amounts of diverse example data to be covered by the algorithm often complicate the process further. The world has seen a surge in data collection and computational power (Sutton et al., 1998). As a result, data for problems have become increasingly easier to acquire and more complex models are viable to develop. Machine learning use algorithms to learn the system by recognizing patterns in data. Suthaharan (2016a) define the machine learning problem as "how to teach a model the system characteristics by using the data to train and validate the model". The ability to robustly solve complex tasks, learning on real-world data and adaptability are key advantages of machine learning (Lison, 2015). Machine learning has a wide range of applications including fraud detection in finance, optimization in manufacturing and medical diagnosis in medicine (Alpaydin, 2009).

2.2.1 Terminology

The data typically consist of multiple examples of scenarios relevant to the problem, called "samples". The samples consist of a series of variables, called "features". An example is a dataset, i.e. set of data, of wine, where each sample represent a distinctive type of wine. Examples of features are pH level, alcohol percentage and citric acid levels. If the sample is linked with the correct classification, or "label", of the wine quality of the sample, the data is "labelled". The sample label could be of multiple "classes", e.x. integers in the range of 1 to 10, for rating the wine quality. Both the input and output is then known. The idea is to train a model that, based on having "seen" different wine samples, their features and their wine quality rating, can predict the rating of an unseen wine sample (Smola and Vishwanathan, 2008). The total amount of samples available are split into a "training set" and a "testing set". The ratio varies, but is typically between 60:40 and 80:20. The model is trained on the training set and evaluated on it's ability to predict the samples in the testing set (Kubat, 2017). The number of samples needed usually grows with the number of features. The training set should cover the problem well by being sufficiently large and diverse (Du and Swamy, 2014).

2.2.2 Learning Methods

Applicable learning methods are determined by the available data. The two main learning methods within machine learning are supervised learning, for labelled data, and unsupervised data, for unlabelled data (Lison, 2015).

Supervised Learning is used for classification, optimization, approximation, signal processing and identification. In supervised learning, the model is adjusted by comparing the predicted value of a sample to its label. The error measure from the comparison is used to guide the learning process (Du and Swamy, 2014). The two main supervised learning methods are classification and regression. Classification is the prediction of correct class given a number of possible classes for a sample not previously seen. The classification is known as "binary", if only two classes consist. In regression, the output value is predicted given a sample (Smola and Vishwanathan, 2008). "Reinforcement learning" is a type of supervised learning, but the exact target value is unknown. The learning method is computationally cheap and minimal human input or understanding is needed (Sutton et al., 1998). The model is punished or rewarded according to how it performs, given an evaluation function (Du and Swamy, 2014). As an example, the evaluation function for an optimization problem might punish the network for each unit of resource or time used.

Unsupervised Learning is used for clustering, feature extraction, signal coding and data analysis (Du and Swamy, 2014). Unsupervised learning does not, contrary to supervised, involve labelled training data (Lison, 2015). Hence the learning is solely based on finding correlations among the training data (Du and Swamy, 2014). As an example, clustering is a method where the data is grouped intro clusters based on their similarities. Semi-supervised learning is used when only a small amount of labelled samples are available. The process of acquiring unlabelled data is typically cheap and automatic, while the labelling process is expensive and manual (Du and Swamy, 2014). During semi-supervised learning, a model is trained using the labelled samples. The model is subsequently used to label the unlabelled samples.

2.2.3 Choice of Learning Method

A decision tree is described in Suthaharan (2016a) to decide on the appropriate learning method for the problem. The decision tree is illustrated in Figure 2.19. The first decision depend on whether the data available for the problem is labelled or not. If the data is not labelled, unsupervised learning methods such as clustering should be employed. Otherwise, the decision tree is further traversed to the characteristics of the domain. The data domain is not dividable if the data is impossible to divide into classes, contrary to the the wine example where the wine is rated using integers between 1 and 10. The appropriate approach would then be to use regression to analyze the trend in the data. If the data domain is divisible, the tree is further traversed to the final decision of the separability of the domain. If the data points associated with the classes are separable, e.g. the features of a distinct wine being contained in its own sample with a corresponding class, the appropriate method is classification of the input space. Using the wine example, a classification of the input space would consist of classifying the wine, i.e. predict the appropriate rating, based on its features. Classification of the feature space involve grouping the features according to the corresponding classes and thus achieve separability. Subsequently, input space classification can be performed.

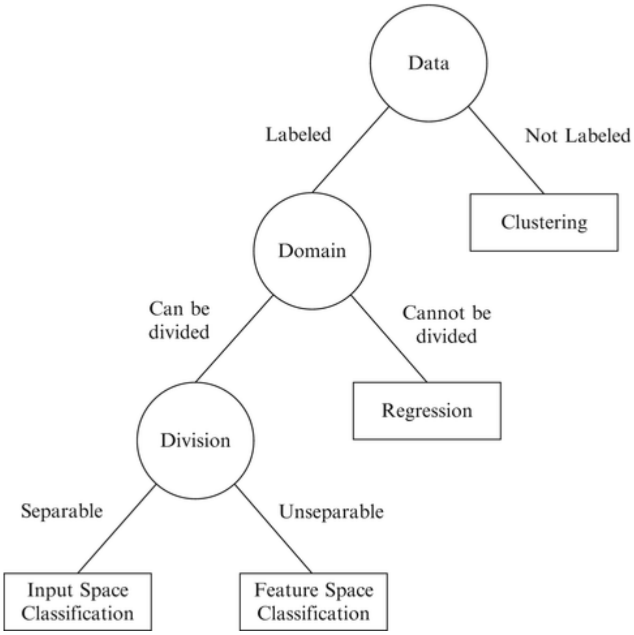


Figure 2.19: Decision tree for deciding machine learning method (Suthaharan, 2016a)

2.2.4 The Development Process

Depending on the choice of learning method, a process for approaching the problem is chosen. The problem in the case study was found to be a classification problem. A process for applying machine learning to real-world classification problems is described in Kotsiantis et. al (2007). The process of developing the optimal classification model is iterative as visualized in Figure 2.20. Different pre-processing methods, algorithms and parameters together result in an extensive number of possible combinations. Multiple combinations should be tested to achieve maximum model performance (Kotsiantis et al., 2007). The following paragraphs in this subsection will introduce the different steps of the process, adhering to the case specifics.

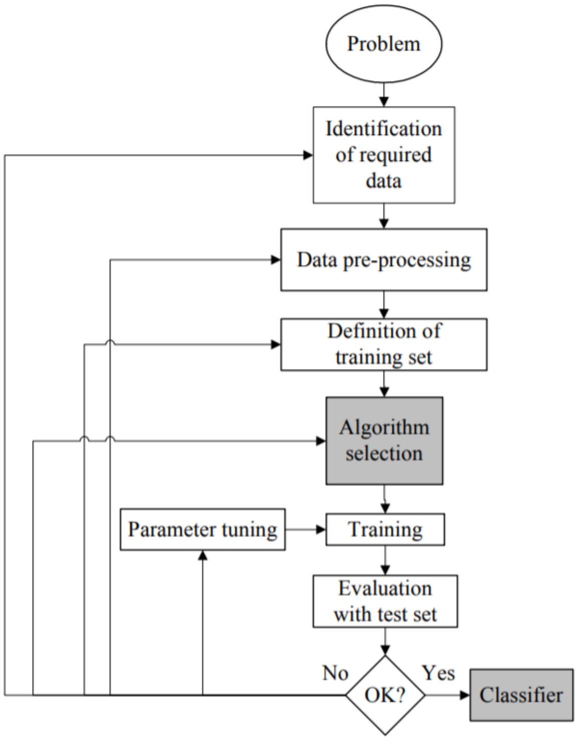


Figure 2.20: Process for applying machine learning (Kotsiantis et al., 2007)

Identification of Required Data and Pre-processing

The first step is to identify and acquire the required data. The data needs to be collected and if an expert is available, expert knowledge should be used to describe the problem and assess the features used from the samples. The following step is to pre-process the data. Pre-processing of the input data is essential to remove irrelevant and/or redundant information to reduce network and computational complexity. The data is typically transformed into a format suitable for machine learning. Feature selection or feature extraction can be used in the pre-processing of the data. When performing feature selection, the features most decisive for distinguishing between the classes are selected and combined into a new dataset. Feature extraction is the process of producing new features from the original features. The performance of the network can improve by reducing the number of features. If the data is uncomplete, e.g. data points lacking for a given interval, one option is to discard the samples. The discarding of samples can be a viable option of the dataset is large, but loss of data in smaller datasets can reduce generalization. Methods to replace the value with a substitute can instead be applied, such as regression (Du and Swamy, 2014).

The Analysis of Variance (ANOVA) F-score was used to rank the features in the case study. ANOVA use variance to rank features according to relevance. The ANOVA analysis is univariate, meaning it does not consider relationships between multiple features (Mwangi et al., 2014).

ANOVA-based feature selection has previously been found to increase classification performance in other studies, including pattern recognition of jet fuels and classification of MRI images (Coutanche et al., 2011; Johnson and Synovec, 2002). The equation for calculating the F-score is found in Chen and Lin (2006). Given data vectors x_k for each sample k, if the number of flaw and normal samples are n_+ and n_-, then the F-score of the ith feature is defined as:

$$F(i) = \frac{(\bar{x_i}^{(+)} - \bar{x}_i)^2 + (\bar{x_i}^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+-1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x_i}^{(+)})^2 + \frac{1}{n_--1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x_i}^{(-)})^2}, \tag{2.1}$$

where $\bar{x}_i$, $\bar{x_i}^{(+)}$ and $\bar{x_i}^{(-)}$ are the averages of the ith feature of the total set, flaw samples and normal samples, respectively. The ith feature of the kth flaw sample is denoted by $\bar{x}_{k,i}^{(+)}$, while the ith feature of the kth normal sample is denoted $\bar{x}_{k,i}^{(-)}$.

Definition of Training Set

The third step is to define the training set and the ratio between the training set and the testing set. A large training set is generally better to cover more scenarios, but can lead to overfitting (Suthaharan, 2016a). "Overfitting" occur if the model is trained too specific to the training data (Lison, 2015). Overfitting can occur due to too many features or samples, or sub-optimal parameters. In the opposite case of having too few features or samples, the model can suffer from "underfitting", where it has not learned enough from the samples (Du and Swamy, 2014).

A dataset is "imbalanced" if the number of samples of each class are not identical. Methods for countering class imbalance are called "sampling". The simplest methods are "random over-sampling" and "random under-sampling". Random over-sampling duplicate random samples of the minority class, while random under-sampling remove random samples from the majority class. Other methods for sampling are typically based on heuristics. Oversampling methods have been found to perform better than undersampling methods, but are prone to overfitting (Batista et al., 2004).

Algorithm Selection and Training

The appropriate algorithm is selected before training the models. The No Free Lunch Theorem suggests that there is no "best" algorithm, and the optimal choice is problem-specific. The theorem is coherent with the conclusions of multiple studies (Lim et al., 2000; KING et al., 1995). The process of choosing the optimal algorithm is thus typically characterized by trial and error, and comparing algorithms is computationally exhaustive. An alternative is to evaluate performances of algorithms on similar problems and data. The general performance of an algorithm can also be found by looking at experimental studies testing a wide range of algorithms on several types of datasets. Experimental results were found in literature relevant to the problem-specifics.

An experimental study conducted in 1995 compared 17 classification algorithms on 12 different datasets (KING et al., 1995). The authors concluded that the choice of algorithm is dependent on the features of the dataset. The association was found by comparing the "skew" and "kurtosis" of the features in each of the datasets to the performance of the algorithms. Skew is a measure of the symmetry, or lack thereof, in the feature distribution. Kurtosis indicate how heavy the tails of the distribution are, compared to a normal distribution. The authors found that decision trees generally performed well on datasets with extreme feature distribution (skew > 1 and kurtosis > 7) and a large portion of binary features (> 38%). Another study was conducted in 2006 as new algorithms had emerged since the aforementioned study (Caruana and Niculescu-Mizil, 2006). 10 algorithms were tested on 11 different binary classification problem. Again was tree-based algorithms found to be top performers. The tree-based algorithm Random Forest was found to perform best among the algorithms not having been subjected to extensive calibration. Furthermore, a study compared the performance of 33 different classification algorithms on 16 different datasets (Lim et al., 2000). 22 tree-based algorithms, 9 statistical algorithms and 2 neural networks were compared. The study concluded that Logistic Regression and Decision trees had the overall best performance. A special type of Logistic regression was the top-performer, but the difference in error rate was negligible and mean training time was 55 times longer than the standard Logistic Regression. The mean training time of the Decision trees and Logistic regression were 5.9 minutes and 4 minutes, respectively.

The Random forest algorithm was chosen for the case study. Random forest is a well known hierarchical approach based on decision trees (Suthaharan, 2016a). A decision tree consist of a single root node connected to other "internal" nodes with outgoing edges. Nodes with only incoming edges are called "leaves". Each internal node split the instance space into multiple sub-spaces using a discrete function (Rokach and Maimon, 2008). A decision tree for binary classification with binary features is illustrated in Figure 2.21. Using the figure as an example, starting at the root node, the value of feature 183 decides the first direction. If the value of feature 183 is 1, the next feature to consider is 489. Following the same logic, the tree is traversed until a class is decided from a leaf node.

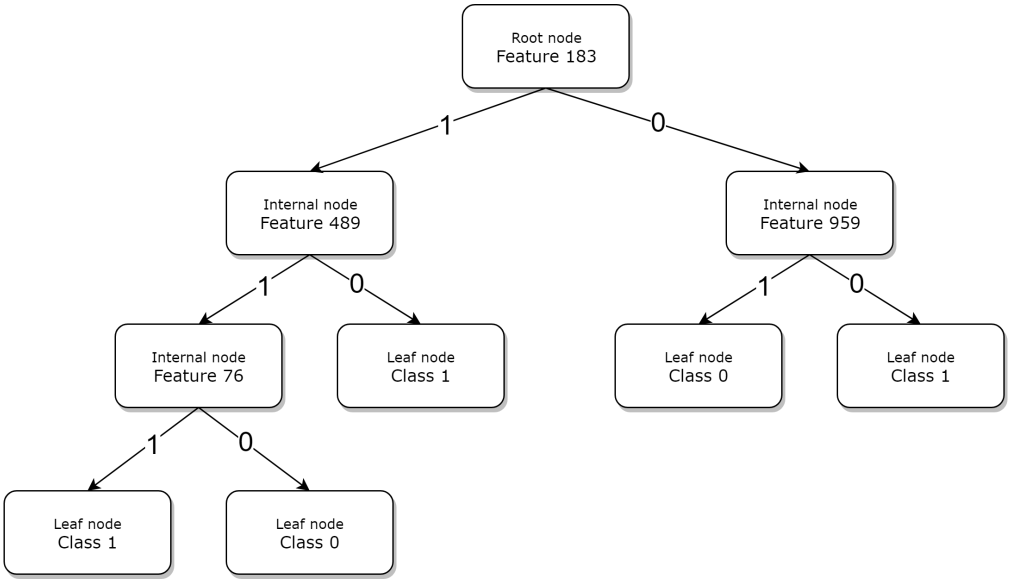


Figure 2.21: Example of decision tree for classification

The Random forest classification algorithm was proposed in Breiman (2001) and defined as "a classifier consisting of a collection of tree-structured classifiers $\{h(x, \Theta_k), k = 1, ...\}$ where the $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input x" (Breiman, 2001). Each tree is constructed by generating a subset K with F number of features from M samples from the total number of training samples, where the values of F and M are random. The features and samples used are not removed from selection for other trees. For each node in the tree, a random feature subset $f < F$ from F is chosen and the features ranked. The feature with the highest information gain within f is used to split the node. The features and samples in subset K can be used more than once. The trees are independent of each other and grown to the largest extent possible (Archer and Kimes, 2008). The randomness used in the constructing of trees, and the large number of trees, make the Random Forest generally robust against overfitting (Lehmann et al., 2007).

The ranking of features in f can be conducted using different measures. The Python library Sci-Kit Learn (Buitinck et al., 2013) was used to implement the algorithm. The default measure used in the Sci-Kit Learn library, the Gini Index, was used in the case study and can be expressed as:

$$\sum \sum_{j \neq i} (f(C_i, T)/|T|)(f(C_j, T)/|T|), \tag{2.2}$$

where $f(C_i, T)/|T|$ is the probability that the selected sample belongs to class C_i (Rodriguez-Galiano et al., 2012). The class of an input sample from the test set is predicted by choosing the class with the highest mean probability estimate across the trees, i.e. the forest.

The number of trees and the number of features considered during the node split selection are the two most decisive parameters to tune, i.e. adjust and test until the optimal value is found, for the Random forest algorithm. The tuning of algorithm parameters are an important part of achieving optimal models (Du and Swamy, 2014).

Evaluation with Test Set

The models are subsequently used to predict the classes of the samples in the test set. The predicted classes are then compared to the true labels of the samples to measure the performance. Different performance measures can be used. The classification problem in the case study was found to be binary, i.e. each sample were to be classified as either of two different classes. The most popular performance measures for binary classification are based on the values of the confusion matrix, namely true positive (TP), false positive (FP), true negative (TN) and false negative (FN) (Sokolova and Lapalme, 2009). The confusion matrix for binary classification is shown in Figure 2.22.

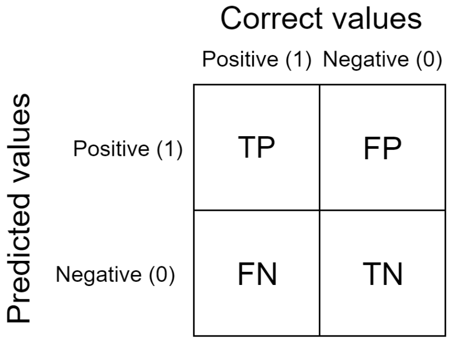


Figure 2.22: Standard confusion matrix

4 different performance measures are based on the confusion matrix values. Accuracy indicate the overall effectiveness of the classifier, see Equation 2.3. Precision is a measure of how precise the classifier is predicting positive values, see Equation 2.4. Recall measure the ability to identify the positive samples. The formula is presented in Equation 2.5. Lastly, the F1 score combine Precision and Recall to calculate the relation between positive labels in the data and those predicted by the classifier, see Equation 2.6. The F1 score provide the harmonic average between Precision and Recall to evaluate classification using a single value. All four performance measures score the performance between 0 and 1, from worst to best, respectively.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.3}$$

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.6}$$

Chapter 3

Case Study

This chapter describes the specifics of the case study and the methods used. The case company is introduced, followed by a description of the rail inspection carried out by the case company and the inspection companies. Furthermore, the process for developing the rail flaw detection models is described in detail.

3.1 Case Company

Bane NOR is a state-owned company responsible for the national railroad infrastructure in Norway. The company has roughly 4 400 employees and maintain 4 200 km of rails (Bane NOR, 2018). 200 000 passengers travel daily on the 1 900 daily train routes. Bane NOR expects an increase in both passengers and freight in the coming years. From 2011 to 2016, the number of train journeys increased from 59.4 million to 74.2 million (Bane NOR, 2017).

In 2017, Bane NOR had a budget of NOK 2.4 billion for renewal projects and NOK 6.3 billion for maintenance. Safety is stated as the primary reason for maintaining the railways. Additionally, sufficient maintenance is stated as a prerequisite for punctual traffic by detecting faults and defects early. The focus on safety is further emphasized through their stated aim "to make sure that no lives are lost" (Bane NOR, 2017). Head of Non-Destructive Testing at Bane NOR, Harald Schjelderup, was the case company contact.

3.2 Inspection Process

The rail inspection process employed by Bane NOR was compiled through interviews with the Bane NOR contact. The rail inspection process is illustrated in Figure 3.1. Bane NOR hires companies specializing in railway inspections on a yearly basis. The two contenders for the contract are Sperry Rail Services from England and the Dutch company EURAILSCOUT. The inspection companies drive a train car that collect ultrasound data. The inspections are generally conducted yearly. On distances where the trains maintain higher speeds, e.g. between the Oslo City Centre and Oslo Airport Gardermoen, the rate of inspection is more frequent. Bane NOR receives B-scans and inspection data for every suspected flaw. A few weeks later, a crew from Bane NOR travels the complete length of the railway to manually check the locations using a portable ultrasound device. The ultrasound device is able to find flaws above 5 millimeters in size. The locations are checked for flaws and the true flaw properties are registered in a system. If the damage is significant, a work order for maintenance is created. The Bane NOR crew takes pictures of the flaws and locations to help the maintenance crew locate the damage later. 2-3 employees conduct manual inspections full-time for approximately 6 months. Bane NOR does not combine the ultrasound method with other methods. Collecting video data of the rails have been tried in the past, but the quality was insufficient. Bane NOR reports having used the ultrasound inspection method for 20-30 years, but are aware of other, newer methods.

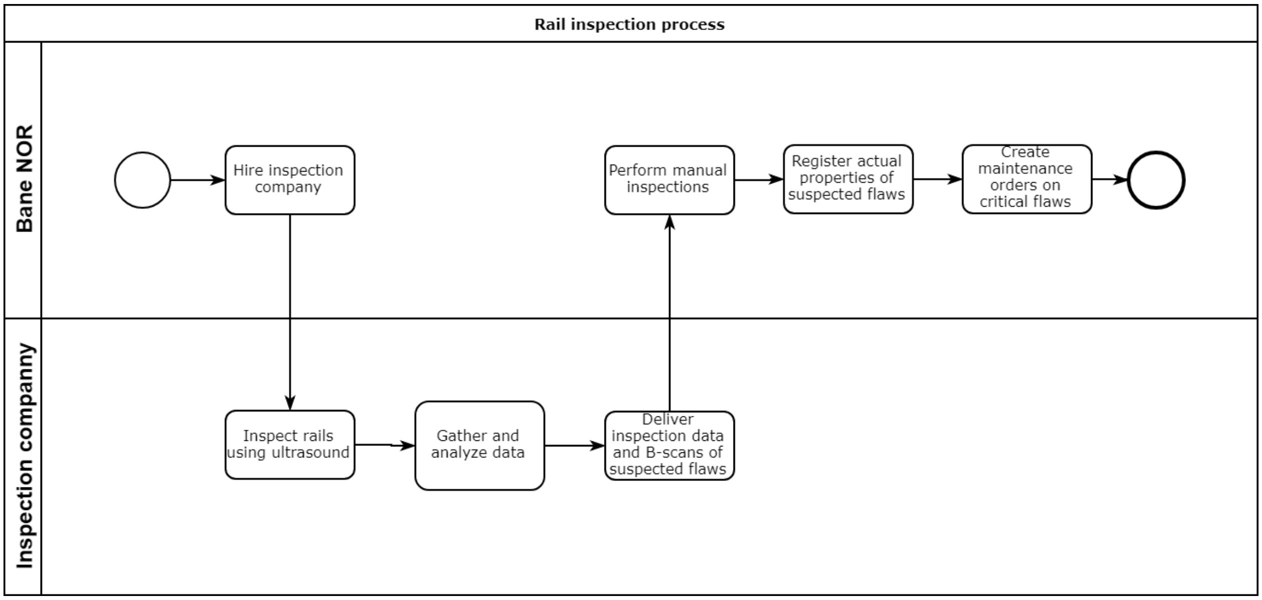

Rail inspection process
Bane NOR
Inspection companny
Hire inspection company
Perform manual inspections
Register actual properties of suspected flaws
Create maintenance orders on critical flaws
Inspect rails using ultrasound
Gather and analyze data
Deliver inspection data and B-scans of suspected flaws

Figure 3.1: The rail inspection process employed by Bane NOR

3.3 Choice of Approach

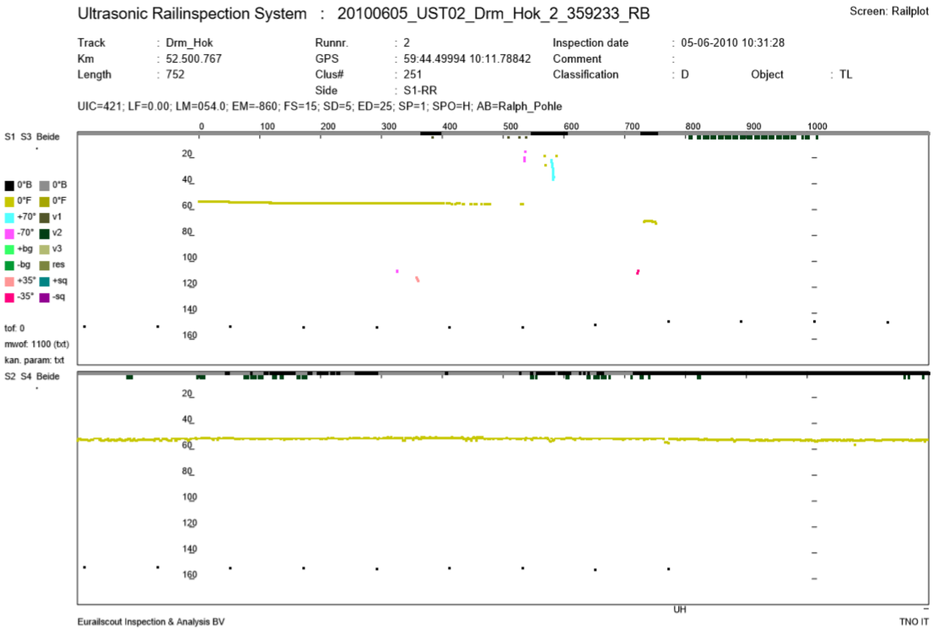
The appropriate machine learning approach was chosen by following the decision tree described in Sutharan (2016a), see Figure 2.19. The decision tree was traversed as follows: The B-scans provided were accompanied by inspection data. 66 properties of each flaw was described. The full list of properties is provided in Appendix B. The data was therefore found to be labelled and clustering was eliminated from selection. Additionally, the domain was found to be divisible. The flaw property used as label for each B-scan was the flaw status, indicating whether the suspected flaw was found, not found or below the threshold for registration. The domain was thus divisible into three categories. Regression was therefore discarded. Finally, the data points associated with the classes were found to be separable, as the features for each sample were split into separate files. The appropriate approach was therefore "input space classification", i.e. classification of the B-scans. Due to classification being chosen, the process described in Kotsiantis et al. (2007) for applying machine learning to real-world classification problems was followed. The process is illustrated in Figure 2.20. Each step is illustrated using figures to help explain the process.

3.4 Subset Selection

The first step in the process of developing the machine learning models was to identify the required data. From theory on ultrasound representations and machine learning, it was known that the B-scans, which are indicative of flaws, and the correct label for each sample, was required for developing supervised classification models.

3.4.1 B-scans

The data was found to be highly diverse. Of the 26 488 files, 2 623 were .pdf document files and the rest .jpg image files. Extracting features from an image, i.e. pixels, were found to be manageable, while extracting the features from document files was deemed too labour intensive. Figure 3.2 depict an example document file. Consequently, the documents were eliminated from further selection.



Ultrasonic Railinspection System : 20100605_UST02_Drm_Hok_2_359233_RB

Screen: Railplot

Track	: Drm_Hok	Runnr.	: 2	Inspection date	: 05-06-2010 10:31:28		
Km	: 52.500.767	GPS	: 59:44.49994 10:11.78842	Comment	:		
Length	: 752	Clus#	: 251	Classification	: D	Object	: TL
		Side	: S1-RR				

UIC=421; LF=0.00; LM=054.0; EM=-860; FS=15; SD=5; ED=25; SP=1; SPO=H; AB=Ralph_Pohle

tof: 0
mwof: 1100 (txt)
kan. param: txt

Eurailscout Inspection & Analysis BV

TNO IT

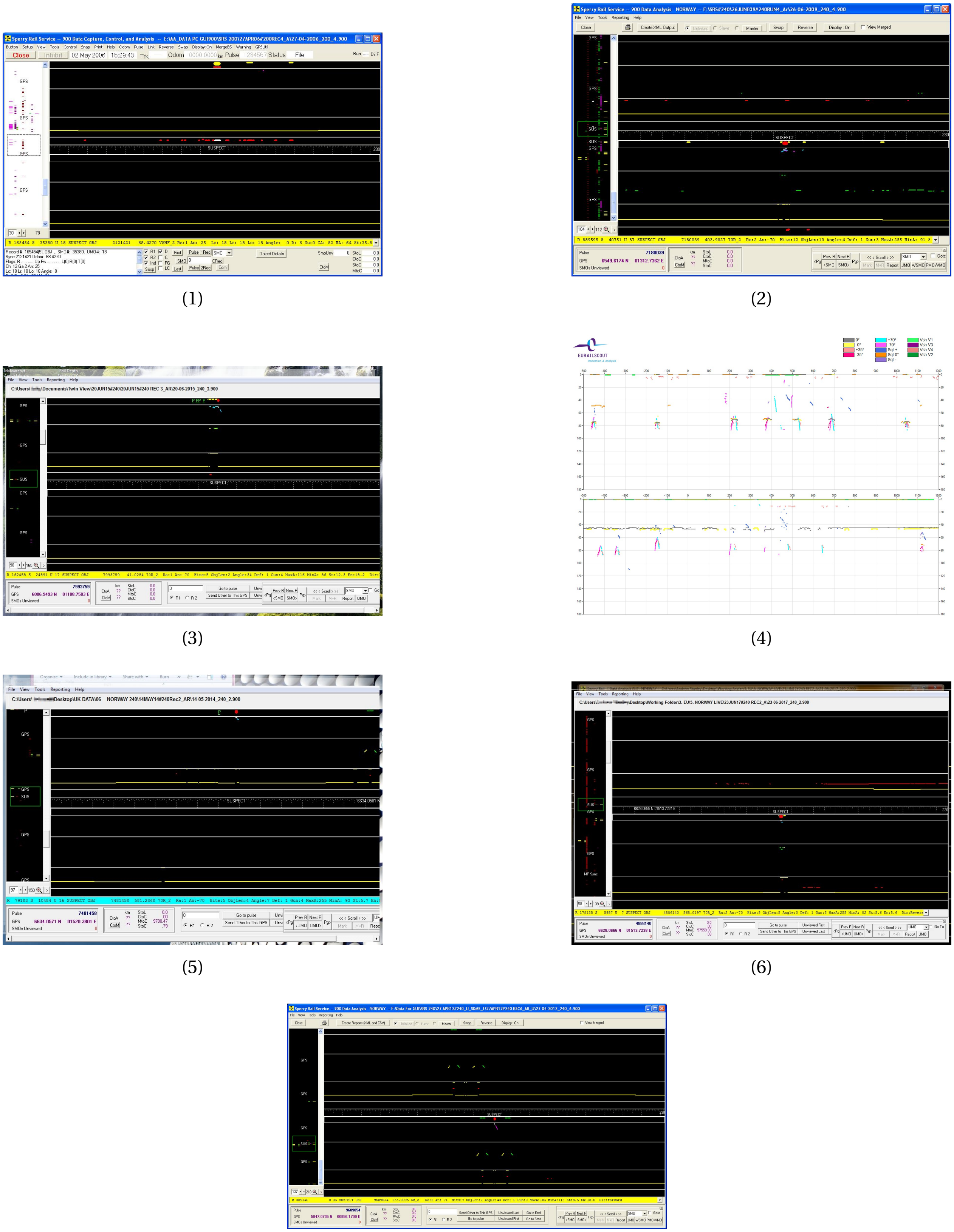
Figure 3.2: Example of B-scan document file from dataset

The image files were of different "resolutions", i.e. image sizes, and formats. Additionally, the files were found to be unsuitable for machine learning in their current formats. The images provided were either "screenshots", i.e. pictures of the computer screen, of a computer program used for inspecting B-scans, or a visual presentation of the B-scans, not the raw B-scans. The different formats can be viewed in Figure 3.3. The diverse data was consistent with Bane NOR's use of different providers over the years.

The training data have to be of identical resolution and format. The image files were inspected to find a subset of an appropriate format and number of samples. The resolutions were found to be specific to the different providers and formats. Code was written in the programming language Python to list the different resolutions and number of samples of each. 267 different resolutions were found, ranging from (678, 596) to (1536, 1136) and from 1 to 2 569 samples. The dataset should be sufficiently large and diverse to cover the problem (Du and Swamy, 2014). As a result, resolutions with less than 1 000 samples were discarded. 7 resolutions remained as presented in Table 3.1. The different formats of the samples are presented in Figure 3.3.

Table 3.1: Resolutions having more than 1000 samples

Number	Resolution	Samples
1	(1024, 657)	2569
2	(1024, 737)	2339
3	(1107, 728)	2029
4	(1536, 1136)	1811
5	(1023, 728)	1377
6	(1206, 837)	1186
7	(1296, 768)	1098



(1)

(2)

(3)

(4)

(5)

(6)

(7)

Figure 3.3: Formats corresponding to resolutions in Table 3.1

Format type 4 in Figure 3.3 from inspection company EURAILSCOUT with resolution (1536, 1136) was chosen due to having the most standardized format, instead of screenshots of a software program. Additionally, despite not having the most samples, the other alternatives had text in the B-scans, potentially creating noise and corrupting the data. The other formats were also found to have horizontal dividing lines in the B-scans of the same colour as the flaw indicators, thereby complicating the process of removing them. Figure 3.4 is an image from the chosen subset.

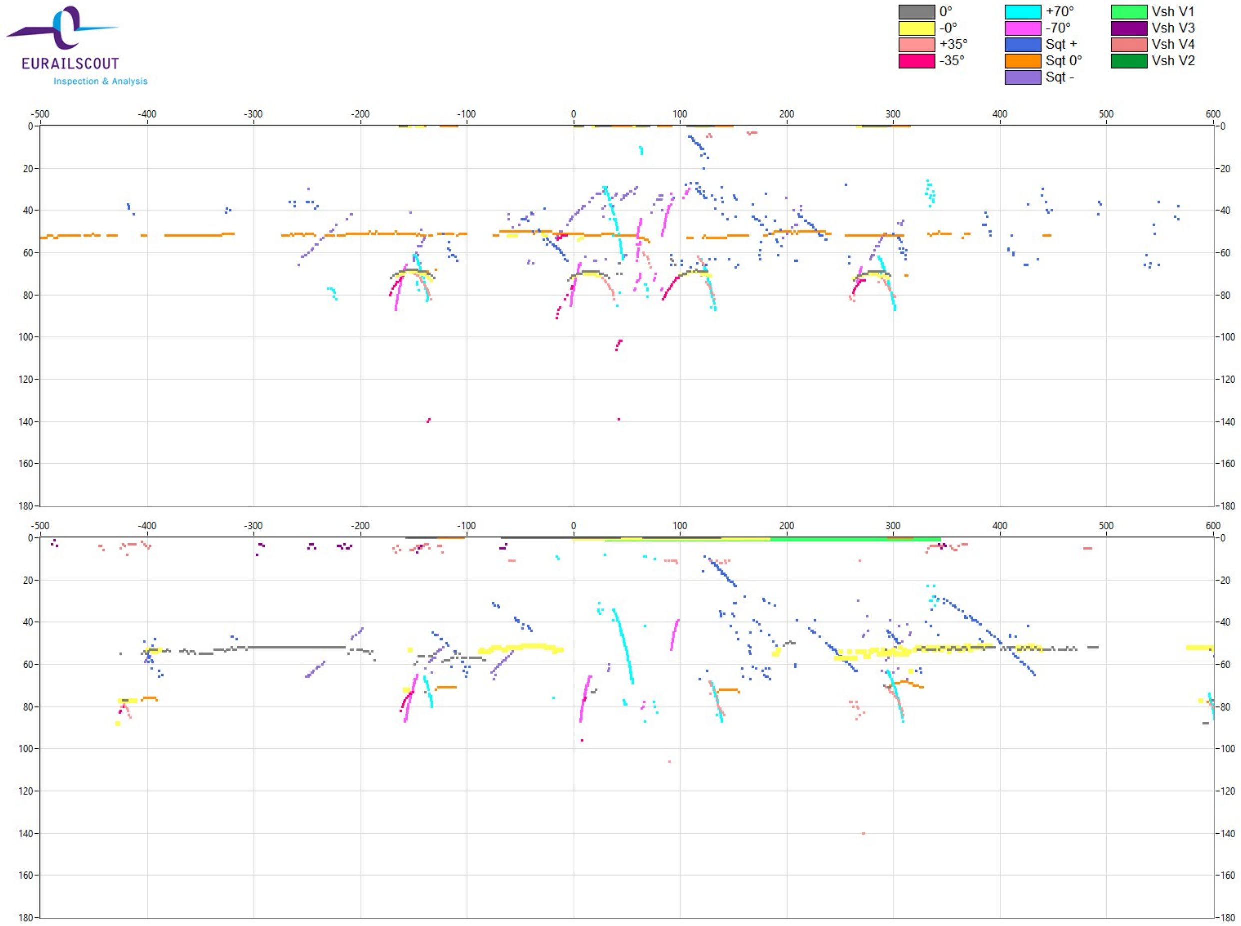


Figure 3.4: Example of visual representation image from subset

Two B-scans were included in the chosen format, one for each rail, as seen in Figure 3.4. The top B-scan is supposed to represent the right side rail, but the inspection companies fail to stay consistent according to the Bane NOR contact. The scales of the axes in the B-scans are presented in millimeter. 13 different colours represented in the B-scans, indicated in the top right of the image. The 3 "Sqt" signals indicate squat flaws, while 4 "VSH" signals indicate VSH flaws. The remaining 6 colours are from the standard ultrasound method of having ultrasound probes in 3 different angles from both sides of the rail to detect flaws.

The ranges of the vertical axis were found to be consistently between 0 and 180, throughout the subset. The minimum value for the horizontal range was largely consistent with -500, while the maximum value ranged from 500 to 1 235 500. The Bane NOR contact reports that they have asked the inspection companies to stay consistent with the ranges. The inconsistency also complicate the task of locating the flaws during manual inspections.

3.4.2 Inspection Data

Bane NOR provided inspection data for each suspected flaw in an Excel file, in addition to the B-scans. The inspection data included properties of the suspected flaws according to the inspection companies, followed by the true flaw properties found by Bane NOR during manual inspection. 66 different properties are registered for each flaw. Examples of properties are the location, priority status and size of the flaws. During manual inspection, flaws are reported as found, not found er below the threshold for registration. The full list of properties, including example data, can be found in Appendix B. Properties, in the full list of properties, starting with the letter S have been registered by the inspection companies, while categories with D were registered during manual inspection. The properties of the flaw found during manual inspection was considered more reliable due to being a more thorough inspection, and therefore considered the true state of the suspected flaw, i.e. the label of the sample. According to the inspection data, the B-scans provided by Bane NOR was from the time interval 2003-2019. The selected subset of B-scans was from 2018. 1 169 of 1 811, or 64.5%, of the samples in the subset are of flaws, if including flaws below threshold as flaws. The percentage of false positives was thus 35.5%. If the flaws below the threshold were considered false positives, the percentage of false positives became 75.8%. The ratio of false positive, i.e. suspected flaws not found, in the subset was found to be nearly twice as high as in the complete set. The percentage of false positives is presented in Table 3.2.

Table 3.2: Percentage of false positives

Classification(s)	**Complete set**	**Subset**
Not found	18.4%	35.5%
Below threshold	20.9%	40.4%
Not found and below threshold	39.3%	75.8%

Conducting manual inspection of flaws below the threshold could be regarded as redundant. However, due to the B-scan representing an actual flaw, although small, the flaws below the threshold were regarded true positives for the purpose of the case study. The classification problem was therefore binary with only two classifications, either flaw or normal. Prior to training the classification models, irrelevant or redundant features must be removed to achieve more effective and efficient training (Yu and Liu, 2004). In addition, the data needs to be transformed into a suitable format for machine learning.

3.5 Pre-Processing

With the subset consisting of image files, the features were represented by pixels. Every pixel has a value for red, green and blue (RGB). Each of the values are between 0 and 255. Each sample, i.e. image, in the subset was to be converted into the matrix form:

$$\begin{bmatrix} x_{00} & x_{01} & x_{02} & \dots & x_{0c} \\ x_{10} & x_{11} & x_{12} & \dots & x_{1c} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{r0} & x_{r1} & x_{r2} & \dots & x_{rc} \end{bmatrix},$$

where x_{rc} represent the feature, i.e. the pixel, in row r and column c. Each sample on the the matrix form was to be further converted into a vector:

$$[x_0, x_1, x_2 \dots x_k], \tag{3.1}$$

where x_k is the feature of the k sample. The target values, i.e. the correct classification of each sample, were also to be converted into a vector:

$$[y_0, y_1, y_2 \dots y_k], \tag{3.2}$$

where y_k is the classification of the kth sample. A total of 6 processing iterations were conducted on each sample to have the samples converted into the necessary matrix format: B-scan extraction, grid removal, sensor subset extraction, colour removal, resolution reduction and binary conversion. Additional subsets were created from the original subset during the pre-processing steps to evaluate the methods later.

3.5.1 B-scan Extraction

The format of the samples delivered to Bane NOR is intended to be easily interpreted. However, only the ultrasound signal data presented as B-scans are relevant for deciding the condition of the rails. Therefore, the B-scans had to be extracted from the current format. The B-scans are located in the regions marked in Figure 3.5.

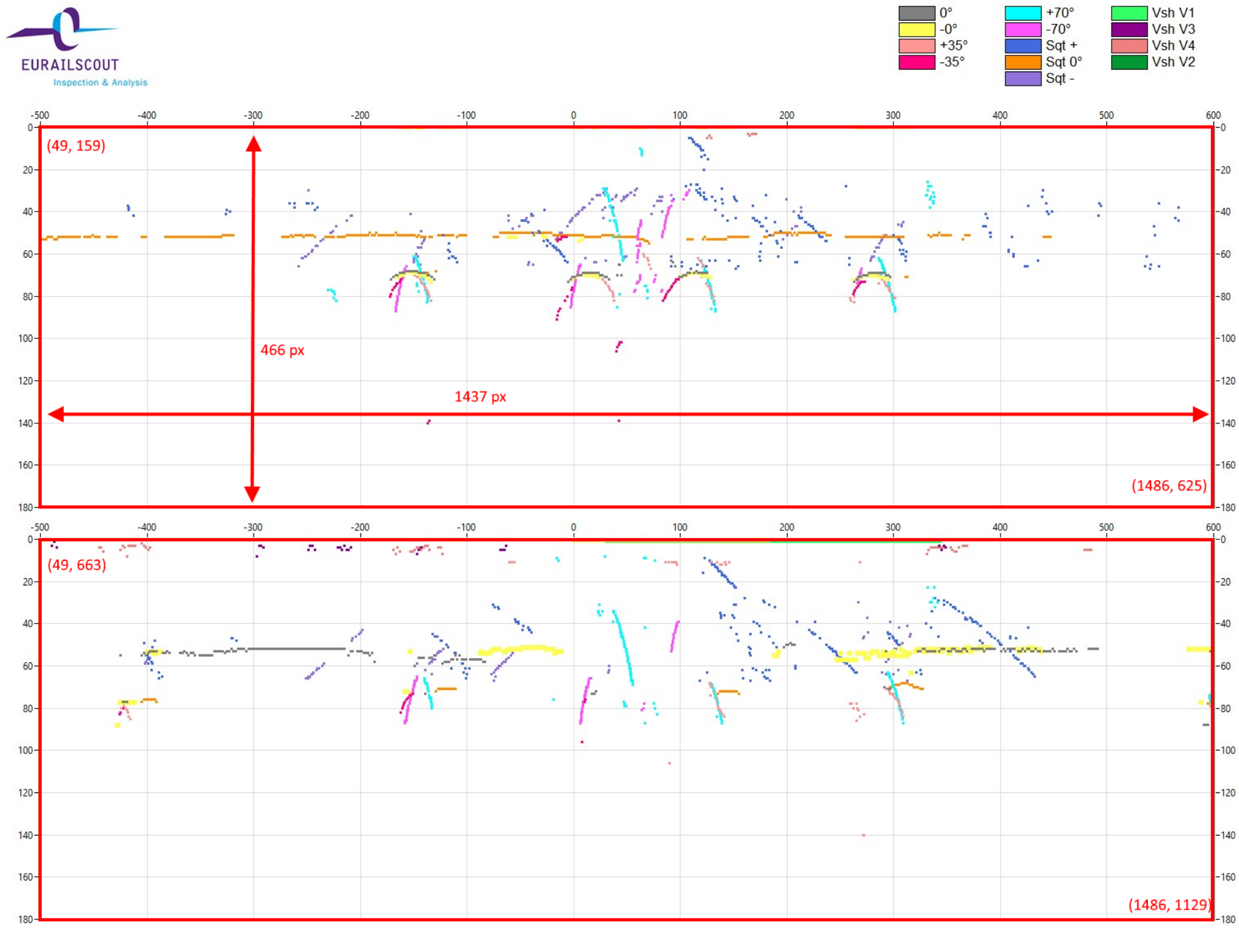


Figure 3.5: Regions of graphs marked with pixel coordinates and size

First, the correct B-scan to extract was chosen for each sample. The inspection data described in which rail the flaw was found. The regions of the two B-scans were found to be at pixel locations (49, 159, 1486, 625) and (49, 663, 1486, 1129), where the first two integers indicate the coordinates (x,y) of the upper left corner and the following two integers indicate the lower right corner of the rectangle-shaped region, as indicated in Figure 3.5. The regions were confirmed to be of equal size. Each B-scan had a dimension of 1437x466 pixels. To minimize the number of features, the range used by the B-scans throughout the subset was found. As an example, if the lower half of the B-scan was never used, i.e. never contained a signal, then half of the features would be redundant. The full horizontal range in the B-scan were found to be used. The highest y value for the upper and lower B-scan was found to be 403 and 404, respectively. The vertical range was set to 0-404 to keep the dimensions consistent. The part of the B-scan outside the range was not extracted as a result. The new B-scan dimension was thus 1437x404. By reducing the vertical range from 466 to 404, the number of features was reduced by 15.3%, or 89 094 features. The resulting extraction of the lower B-scan in Figure 3.5 is depicted in Figure 3.6.

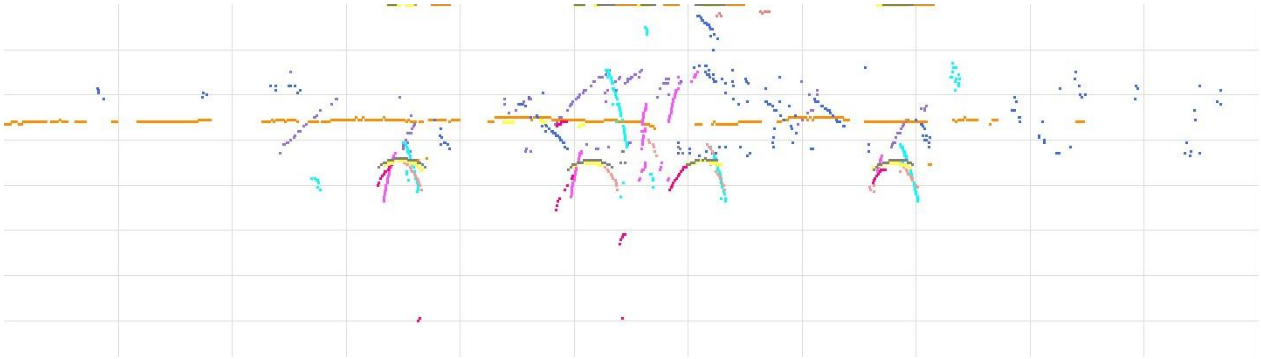


Figure 3.6: Extracted B-scan from image file

3.5.2 Grid Removal

The grid was removed from the B-scans as it was not relevant to the indication of flaws. The grid could however create noise for the model, by introducing redundant or misleading features. To remove the grids, the RGB colour values of the pixels were extracted. The RGB values of the grid were found to all be above 200. Subsequently, all pixels with RGB values above 200 were converted to white. Signal data colliding with the grid was found to have been placed on top of the grid and was therefore not lost in the process. The B-scan in Figure 3.6 without grid can be seen in Figure 3.7.

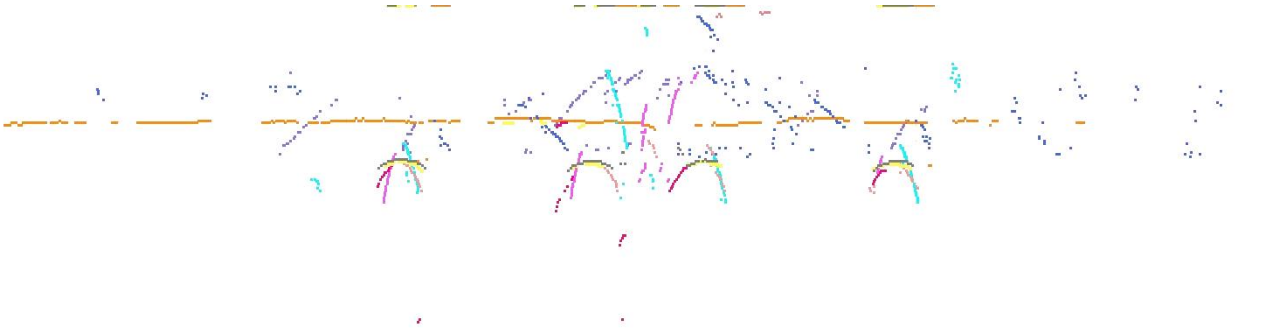


Figure 3.7: Grid removed from B-scan

3.5.3 Sensor Subset Extraction

The aforementioned 13 different colours represented in the B-scans originate from signals of three different sensor methods to detect different types of flaws: standard, squats and VSH. The colour indications can be seen in Figure 3.8. Three additional subsets were created by only including the colours for each method.

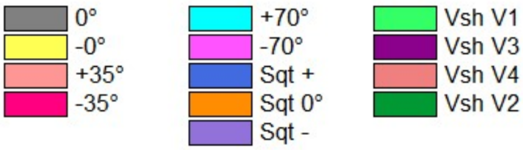


Figure 3.8: Colour indications for the B-scans

The colours used in the B-scans were found to vary from the declared colour indications. The B-scan without grid in Figure 3.7 was found to contain 8 428 different colours, not 13 as indicated. Almost no pixels were found by choosing the pixels with the RGB values of the colour indications. Therefore, a threshold was set to include colours slightly diverting from the original colour indication. A threshold value of 50 was found to include most of the desired colours for each method while also excluding the unwanted. The following RGB values were thus matched for each colour indication:

$$(R, G, B) \begin{cases} R \in [R - 50, R + 50] \\ G \in [G - 50, G + 50] \\ B \in [B - 50, B + 50] \end{cases} \tag{3.3}$$

where R, G, and B are the RGB values of the colour indication. Figure 3.9 illustrate examples of colours, with their respective RGB values, within the range of the yellow colour indicator in Figure 3.8. The colour indicator is situated in the middle of the figure for reference.

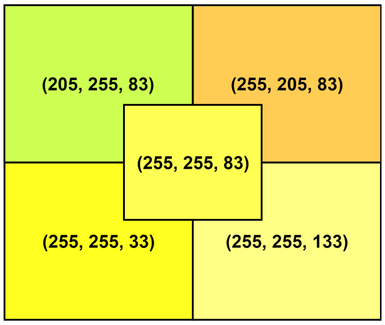
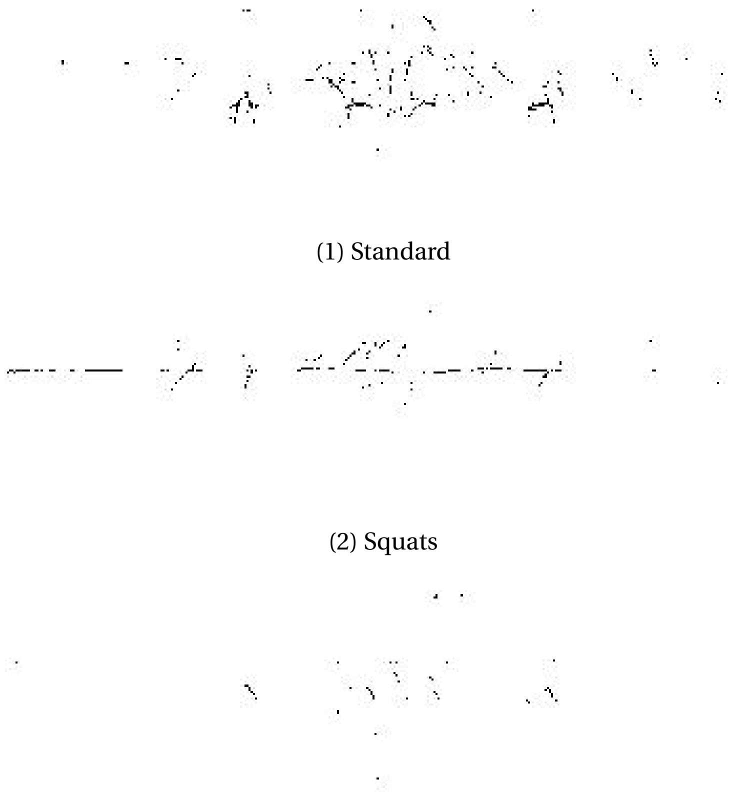


Figure 3.9: Range of colours for the yellow colour indication in Figure 3.8

The three versions of the B-scan from Figure 3.7 are displayed in Figure 3.10. The pattern found in the Squat subset was coherent with squats as surface flaws appear in the horizontal direction of the B-scan. The subset from sensors detecting VSH showed signal data with a dominant vertical direction, further validating the choice of threshold value.



(1) Standard

(2) Squats

(3) VSH

Figure 3.10: Signals for each method extracted from Figure 3.7

3.5.4 Colour Removal

Splitting the subsets based on colour to produce one subset for each colour indicator was deemed unfavorable as most samples would have little to no signal data left. The colours in the B-scans were subsequently converted to black and white. A B-scan after black and white conversion can be seen in Figure 3.11.

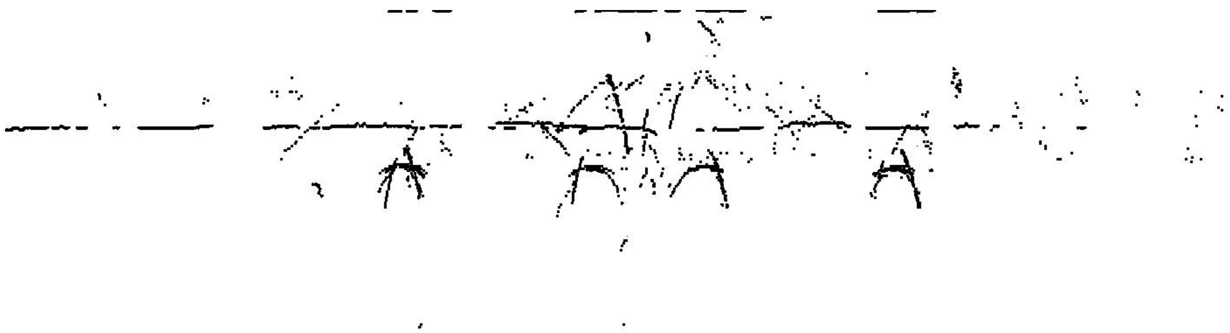
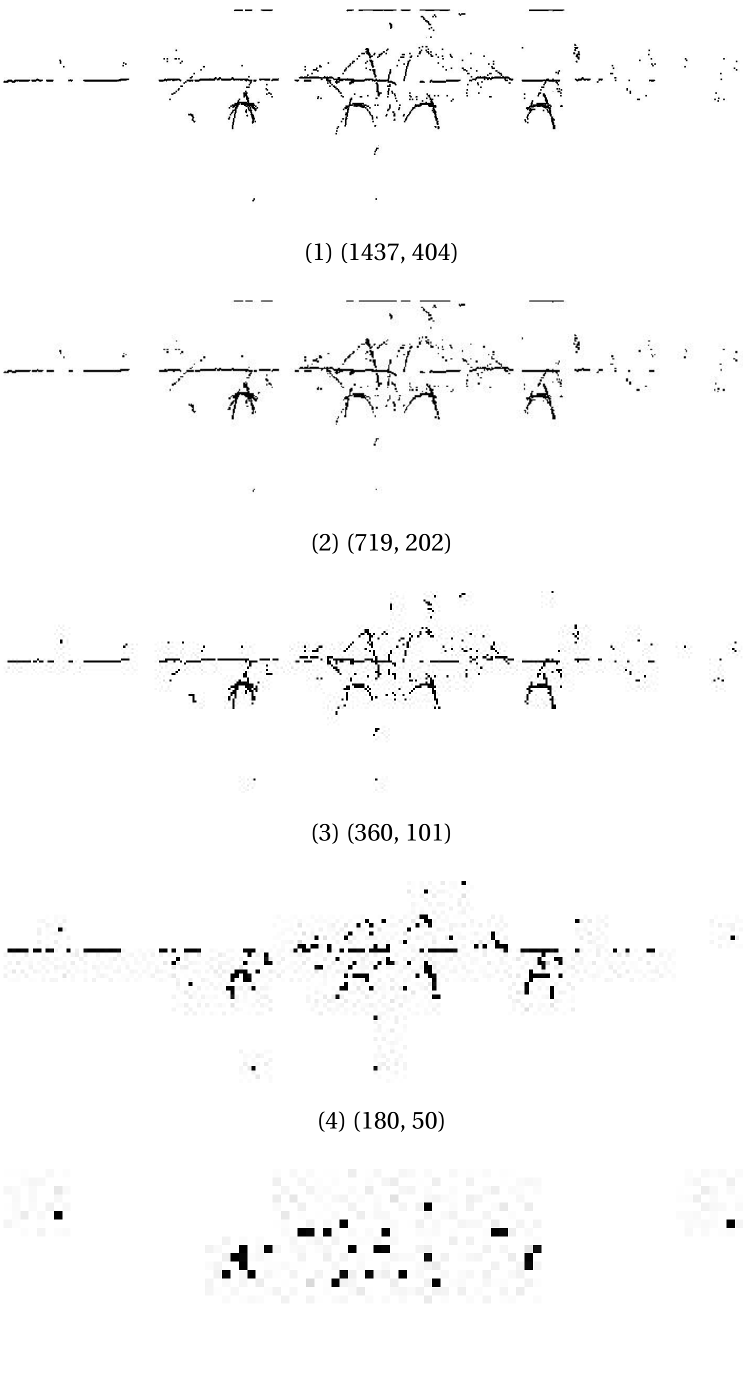


Figure 3.11: B-scan converted to black and white

3.5.5 Resolution Reduction

Each sample consisted of 580 548 features due to resolutions of 1437x404. By reducing the resolution, the number of pixels, and thus features, is reduced. The resolution was halved until the pattern was no longer distinguishable. The resolution was lowered by choosing the nearest pixel from the input image for the output image. As a result of the majority of the pixels being white, the B-scans became increasingly white. The aim was to remove irrelevant or redundant features, without losing information. Figure 3.12 illustrate the incremental reduction of the resolution. Resolution (180, 50) with 9 000 features was chosen as the maximum decrease in resolution due to being the last iteration where the pattern was clearly distinguishable.



(1) (1437, 404)

(2) (719, 202)

(3) (360, 101)

(4) (180, 50)

(5) (90, 25)

Figure 3.12: Incremental resolution reduction of B-scans

3.5.6 Feature Selection

Feature selection was conducted as an alternative to resolution reduction. By removing features with no variance, i.e. features having the identical value in all samples, the number of features were reduced to 471 602 or a reduction of approximately 18.8%. The number of features was further reduced by calculating the ANOVA F-score of each feature and only select the features with the highest F-score. Figure 3.13 showcase the locations of the 50 000 best features found in the B-scans, approximately 1/10 of the total number of features, according to the ANOVA F-score. The selected best features are black, while the discarded features are white. A black border was added to distinguish the size of the B-scan. The lower half of the features can be seen to be mostly discarded.

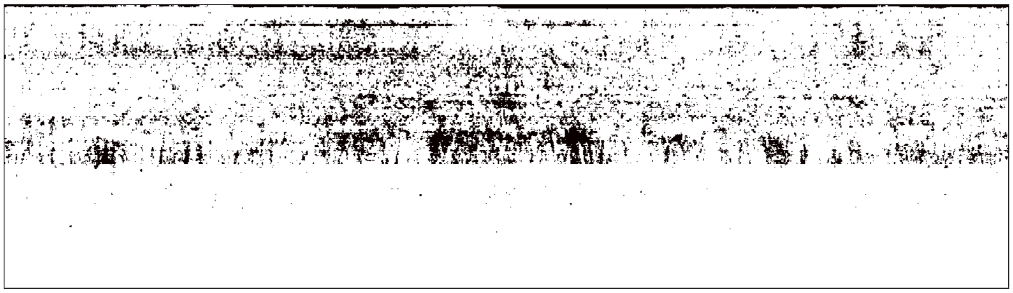
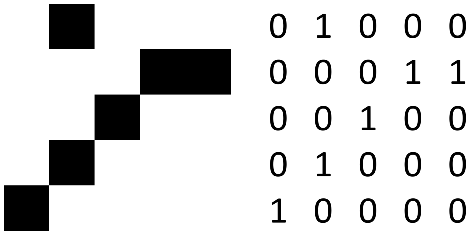


Figure 3.13: The 50 000 best features from the B-scans

3.5.7 Binary Conversion

In the final stage of the pre-processing, the features were converted to binary form. A matrix with the same dimensions as the B-scans was created for each sample and filled with 0's. If the pixel in the black and white converted B-scan was black, the corresponding location in the matrix was changed to 1. The process is illustrated in Figure 3.14, where the 0's correspond to the white areas and the 1's to the black areas. The matrix's was then converted to the required form of Equation 3.1.



$$\begin{matrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{matrix}$$

(a) Before (b) After

Figure 3.14: Black and white pixels converted to matrix of binary values

Figures 3.15 and 3.16 visualize the vectors for features and classes, respectively. Both vectors consist of binary values, but there is only one classes vector, while each sample is represented by a features vector. The first value being 1 in the features vector indicate that the first feature in the B-scan , i.e. the first pixel, is representing a signal. By the same logic, the third feature does not represent a signal. For the classes vector, the first value indicate the label of the first sample. As seen in the figure, the first sample has label 1, indicating a flaw. The third sample indicate a normal condition.

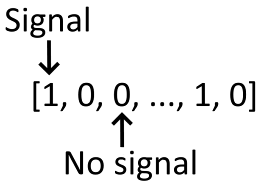


Figure 3.15: Features vector

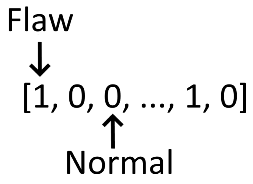


Figure 3.16: Classes vector

3.6 Algorithm Selection

Because the selection is problem-specific, potential algorithms were evaluated by their performance on similar problems in literature. Because no literature has been produced on the exact same data and with A-scans being significantly different, experimental comparisons of a wide range of algorithms on multiple datasets were evaluated. The guidelines found in literature was presented in the theory chapter.

The skew and kurtosis of the B-scan samples were found to be 141 and 12, respectively. The high values of kurtosis and skew was caused by the vast amount of 0's compared to 1, in combination with the feature set consisting of only two different values. Figure 3.17 illustrate the distribution of the features. Furthermore, all the features in the dataset were binary. Decision trees were therefore deemed the most fitting algorithm group for the B-scan dataset.

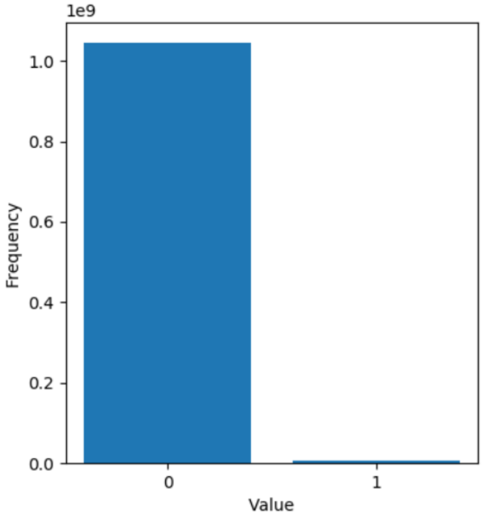


Figure 3.17: Feature distribution in dataset

Tree-based algorithms, i.e. algorithms based on decision trees, were deemed best suited for this problem due to overall good performance on datasets with similar characteristics, combined with low mean training time. The tree-based Random Forest algorithm was chosen due to generally satisfying performance without extensive calibration.

3.7 Training

Training was conducted on a portable laptop with processor Intel i7-7700HQ @ 2.80GHz. The Python library Sci-Kit Learn (Buitinck et al., 2013) was used to implement the algorithm. Different combinations of subsets and parameters were trained to find the optimal combination. Each combination was trained three times to evaluate the performance due to the randomness in the Random Forest. As a result, three models were trained for each combination. The average performance of each combination was calculated. The training sequences are illustrated in Figure 3.18. The highest performing subset was used in the next training iteration, e.g. the K value with the highest performance score was used when testing the number of trees. Up until the parameter tuning, the models were trained with the library default value of 100 trees.

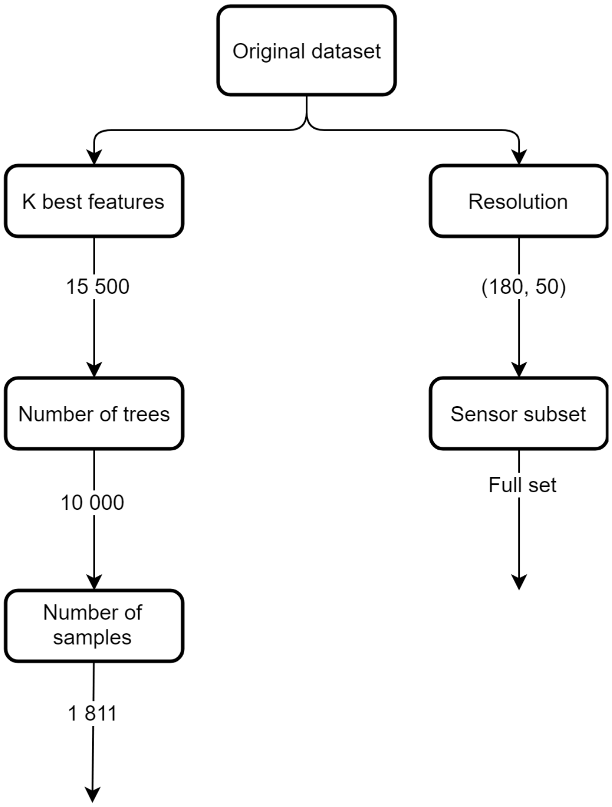


Figure 3.18: The training paradigm

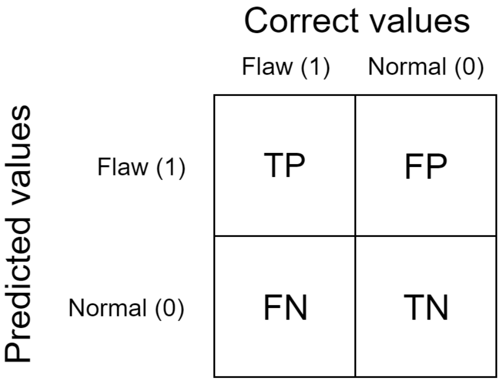
The samples were randomly divided into a training set and a testing set with a ratio of 80:20. The ratio is widely used because it provides a decent portion of the data for training, while still ensuring generalization (Suthaharan, 2016b). The choice of samples for the training set and the testing set was consistent throughout the case study.

Two parameters are commonly tuned for the Random forest classification algorithm; the number of features considered during the node split and the number of trees. The number of trees were chosen as the parameter to tune due to being the most decisive for performance and to narrow the scope. The default number of features considered during the node split selection was set to the square root of the total number of features in the library.

K values 1 000, 10 000 and 100 000 were initially tested to find the K best features. Further testing was conducted with different K values in an attempt to converge on the optimal value. In ascending order, the K values tested were 1 000, 10 000, 13 500, 15 500, 17 500, 25 000, 50 000, 75 000 and 100 000. Next, the optimal parameter value for number of trees was found, by first testing values 10, 100, 1 000, 10 000 and then 5 000, 7 500 and 15 000 to converge on the optimal value. The following training iteration was performed to find the ideal number of samples. The total number of samples were halved for each time and resulted in 4 datasets with 1 811, 906, 453 and 226 samples, respectively. Resolution reduction was tested by halving the resolution each time, resulting in resolutions (1437, 404), (719, 202), (360, 101) and (180, 50). The resolution reduction was found to perform poorly compared to K best features and was therefore not used to find optimal parameter values and number of samples. However, the sensor subsets were tested using the best performing resolution. The resolution was used because the selection of K best features is incompatible with testing the different sensor subsets, as the features associated with the sensor subsets were already chosen. The training set consisted of 939 samples of flaws and 510 samples of normal conditions, or 64.8% and 35.2%, respectively. 81 models were trained in total, on 27 different combinations.

3.8 Evaluation

Following training, the model was tasked with predicting the class of each sample in the test set. The test set consisted of 230 samples of flaws and 134 samples of normal conditions, or 63.2% and 36.8%, respectively. The predicted classes were compared to the true classes. The problem was a binary classification problem and the case-specific confusion matrix is illustrated in Figure 3.19.



	Flaw (1)	Normal (0)
Flaw (1)	TP	FP
Normal (0)	FN	TN

Figure 3.19: Confusion matrix for the case

Scores for true positives (TP), false positives (FP), true negatives (TN), false negatives (FN), Accuracy, Precision, Recall, F1 and run time was calculated to produce results. However, the scores for Rrecision, Recall and F1 were found to produce misleading scores. Upon reviewing the scores, the scores were found to be inaccurate due to the class imbalance of the dataset. The models received favorable scores by predicting nearly all samples to be positive, i.e. flaws. With a class imbalance of 7:3, where positives make up the 70%, the Precision score would be 0.700, the maximum Recall score 1.000 and the resulting F1 score equal to 0.824. Safety is explicitly stated as the main goal for Bane NOR. As a result, the equations were changed to reward models with low ratio of false negatives, i.e. true flaws that will not be inspected, compared to the number of true negatives. The revised Recall equation penalized models with high numbers of false positives. The equations were altered by replacing positives with negatives, e.g. replacing true positives with true negatives, and vice versa. The F1 equation was left original. The revised equations for Precision and Recall was:

$$Precision = \frac{TN}{TN + FN} \tag{3.4}$$

$$Recall = \frac{TN}{TN + FP} \tag{3.5}$$

In addition to the average performance of each combination, results were produced by comparing the top performing models to the current solution. The scores were further used to calculate the prediction accuracy of each of the two rail conditions, flaw and normal.

The optimal model can be chosen by calculating the utility value for each. A possible factor for calculating the utility is cost estimates. The cost of inspecting a suspected flaw can be used. As the manual inspection team travel the complete length regardless of number of samples, the fixed costs must be disregarded to exclusively consider the cost of stopping and inspecting one extra flaw. Additionally, the cost of not inspecting an actual flaw is needed. Different scenarios are presented due to the lack of cost estimates and definite utility values. The function used for calculating the utility U of model k for scenario i is:

$$U(k) = p(TN) * u_i(TN) + p(FN) * u_i(FN), \tag{3.6}$$

where $p(TN)$ and $p(FN)$ are the percentages of TN and FN, respectively, taken from Table 4.7. Furthermore, $u_i(TN)$ and $u_i(FN)$ are the utility values of TN and FN for scenario i, respectively. Utility value $u_i(TN)$ thus indicate the value of not conducting a redundant inspection, whereas $u_i(FN)$ indicate the value of not conducting inspection on a true flaw. The utility of leaving a true flaw unchecked is believed to be negative, while the utility of saving redundant inspections is regarded positive. Three scenarios are envisioned to discuss the choice of model:

1. $u_1(TN) = 1, u_1(FN) = -5$
2. $u_2(TN) = 1, u_2(FN) = -1$
3. $u_3(TN) = 5, u_3(FN) = -1$

Chapter 4

Results

The results from the case study is presented in this chapter. The average performance scores of three models for each version tested is presented using tables and graphs. The graphs are produced to improve readability of the results. The full list of performance scores for each model is available in Appendix C. Furthermore, the top three performing models are compared to the current solution. The results are based off 81 models in total and the duration times are listed in seconds.

4.1 K Best Features

The average performances of the different K best features is depicted in Table 4.1. 15 500 features produced the highest average F1 and Precision score. 50 000 features produced the highest average Recall score. The duration time increased in correlation with the number of features.

Table 4.1: Average performance of different K best features

K	TP	FP	TN	FN	Precision	Recall	F1	Duration
1 000	0.539	0.267	0.099	0.095	0.512	0.271	0.354	0.373
10 000	0.563	0.238	0.129	0.071	0.646	0.351	0.455	1.685
13 500	0.567	0.236	0.130	0.067	0.660	0.356	0.463	3.541
15 500	0.575	0.236	0.130	0.059	0.690	0.356	0.469	3.521
17 500	0.563	0.236	0.130	0.071	0.648	0.356	0.460	2.443
25 000	0.561	0.239	0.128	0.073	0.639	0.348	0.451	3.319
50 000	0.556	0.233	0.133	0.077	0.634	0.363	0.462	5.031
75 000	0.556	0.245	0.121	0.078	0.609	0.331	0.428	6.658
100 000	0.554	0.253	0.114	0.080	0.588	0.311	0.406	8.148

Figure 4.1 visualize the average performances of different K best features. Scores for K values

above 25 000 are not included in the plot to improve readability of scores in close proximity to the highest scores. The score for F1, Precision and Recall does not correlate with the number of features.

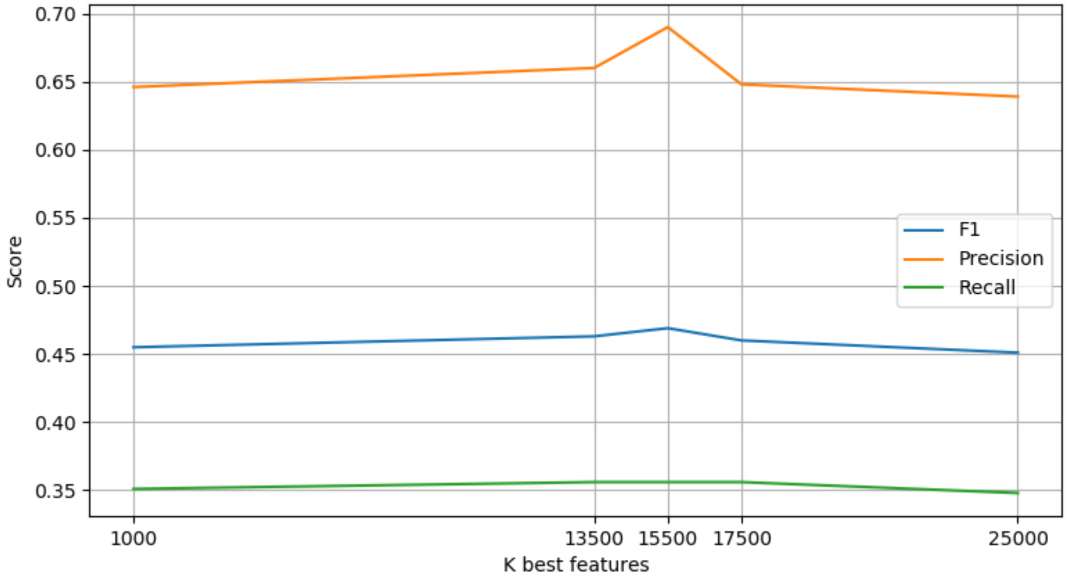


Figure 4.1: Plot of average performances of different k best features

4.2 Number of Trees

The average performances of the different number of trees is depicted in Table 4.2. 10 000 trees produced the highest average F1 score. 15 000 trees produced the highest average Precision score. 10 trees produced the highest average Recall score. The duration time increased in correlation with the number of trees.

Table 4.2: Average performance of different number of trees

Trees	**TP**	**FP**	**TN**	**FN**	**Precision**	**Recall**	**F1**	**Duration**
10	0.515	0.221	0.145	0.118	0.550	0.396	0.460	0.690
100	0.575	0.236	0.130	0.059	0.690	0.356	0.469	3.521
1 000	0.583	0.235	0.131	0.051	0.722	0.358	0.479	36.772
5 000	0.586	0.235	0.131	0.048	0.733	0.358	0.481	164.196
7 500	0.586	0.238	0.129	0.048	0.729	0.351	0.474	274.624
10 000	0.586	0.234	0.132	0.048	0.735	0.361	0.484	399.980
15 000	0.588	0.236	0.130	0.046	0.740	0.356	0.481	506.054

Figure 4.2 visualize the average performances of different number of trees. Scores for number of trees below 1000 were not included in the plot to improve readability of scores in close proximity to the highest scores.

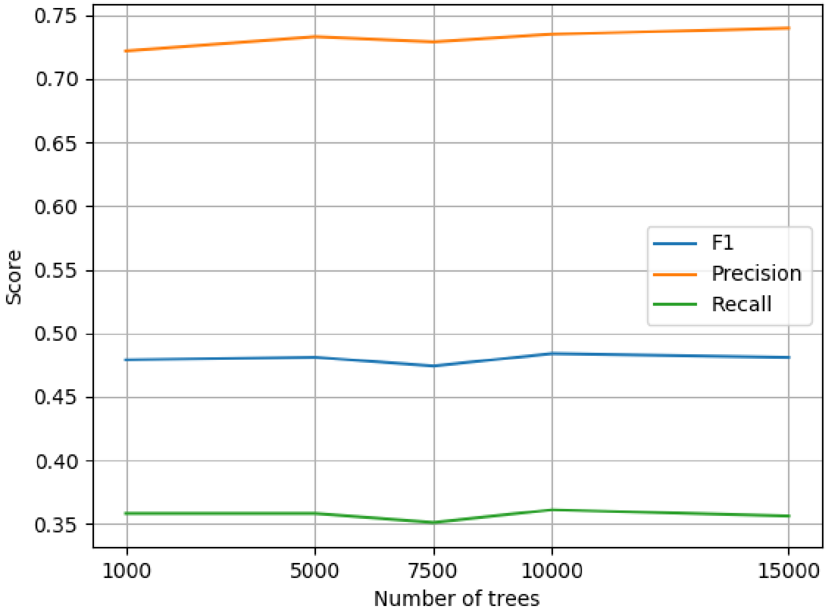


Figure 4.2: Plot of average performances of different number of trees

4.3 Number of Samples

The average performances of the different number of samples is depicted in Table 4.3. 1 811 samples produced the highest average F1, Precision and Recall score. The duration time increased in correlation with the number of samples.

Table 4.3: Average performance of different number of samples

Samples	TP	FP	TN	FN	Precision	Recall	F1	Duration
226	0.622	0.256	0.067	0.056	0.545	0.207	0.300	27.449
453	0.598	0.276	0.080	0.046	0.633	0.224	0.331	78.638
906	0.563	0.263	0.115	0.059	0.660	0.304	0.416	135.185
1 811	0.586	0.234	0.132	0.048	0.735	0.361	0.484	399.980

Figure 4.3 visualize the average performances of different number of samples. All three scores increase in correlation with the number of samples.

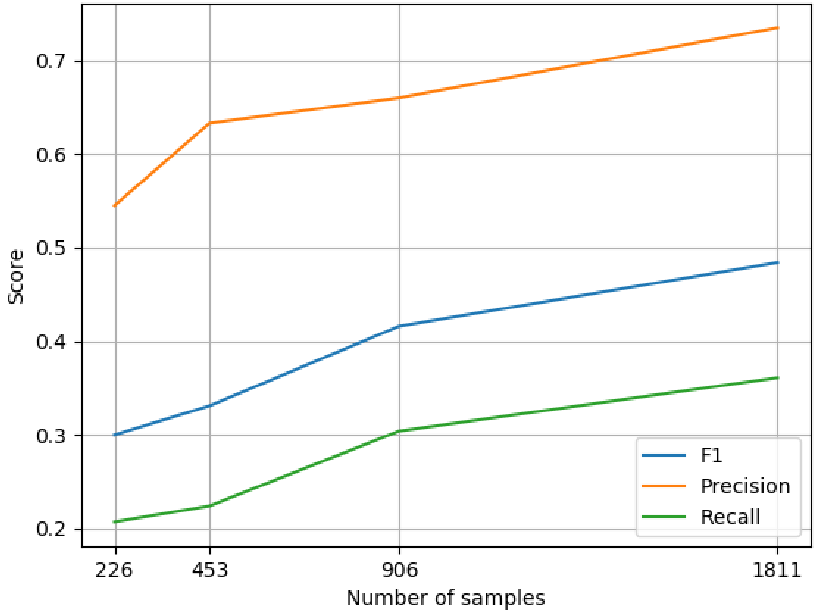


Figure 4.3: Plot of average performances of different number of samples

4.4 Resolution

The average performances of the different resolutions is depicted in Table 4.4. Resolution (180, 50) produced the highest average F1, Precision and Recall score. The duration time increased in correlation with the resolution.

Table 4.4: Average performance of different resolutions

Resolution	TP	FP	TN	FN	Precision	Recall	F1	Duration
(180, 50)	0.498	0.221	0.145	0.136	0.516	0.396	0.448	6.766
(360, 101)	0.508	0.240	0.127	0.126	0.502	0.346	0.410	15.742
(719, 202)	0.506	0.240	0.127	0.128	0.498	0.346	0.408	39.124
(1437, 404)	0.500	0.241	0.126	0.134	0.485	0.343	0.402	120.606

Figure 4.4 visualize the average performances of different resolutions. The decrease in all three scores is correlated with the increase in dimension.

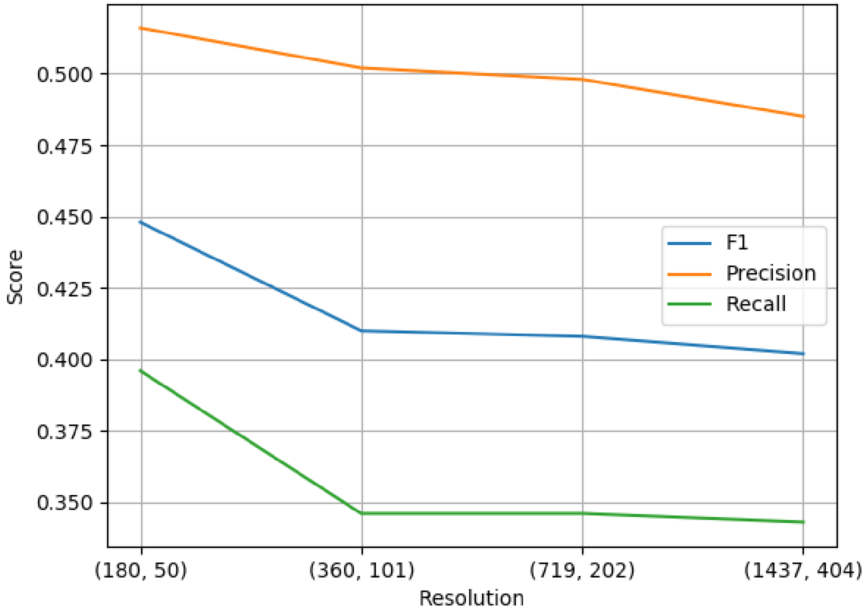


Figure 4.4: Plot of average performances of different resolutions

4.5 Sensor Subset

The average performances of the different sensor subsets is depicted in Table 4.5. The full set produced the highest average F1 score. The full set and Standard subset is tied for the highest average Recall score. The VSH subset produced the highest average Precision score.

Table 4.5: Average performance of different sensor subsets

Sensor subset	TP	FP	TN	FN	Precision	Recall	F1	Duration
Full	0.498	0.221	0.145	0.136	0.516	0.396	0.448	6.766
Standard	0.482	0.221	0.145	0.152	0.489	0.396	0.438	4.094
Squats	0.551	0.277	0.089	0.083	0.520	0.243	0.331	3.035
VSH	0.575	0.281	0.085	0.059	0.593	0.233	0.335	3.178

Figure 4.5 visualize the average performances of different sensor subsets. The three scores are in closest proximity to each other with the Standard subset. Compared to the Standard subset, the Precision and Recall scores of Squats and VSH subset are negatively correlated.

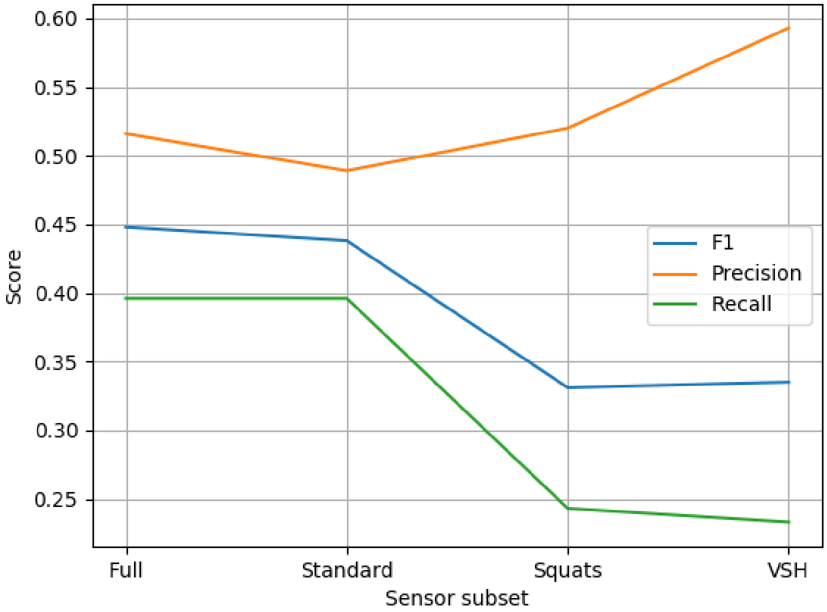


Figure 4.5: Plot of average performances of different sensor subsets

4.6 Top Performances

The models with the highest scores on F1, Precision and Recall is depicted in Table 4.6. The F1 scores are largely consistent, while the recall and precision vary. Models 36 and 37 have similar scores for Precision, Recall and F1, while Model 30 is an outlier.

Table 4.6: Top single performances

Model	Top score	TP	FP	TN	FN	Precision	Recall	F1	Duration
36	F1	0.584	0.229	0.138	0.050	0.735	0.376	0.498	38.744
37	Precision	0.590	0.231	0.135	0.044	0.754	0.368	0.495	159.590
30	Recall	0.512	0.207	0.160	0.121	0.569	0.436	0.494	0.651

The F1, Precision and Recall scores for the top performances is depicted in Figure 4.6. Model 37 improve Precision at the expense of Recall, compared to Model 36. The opposite is true for Model 30.

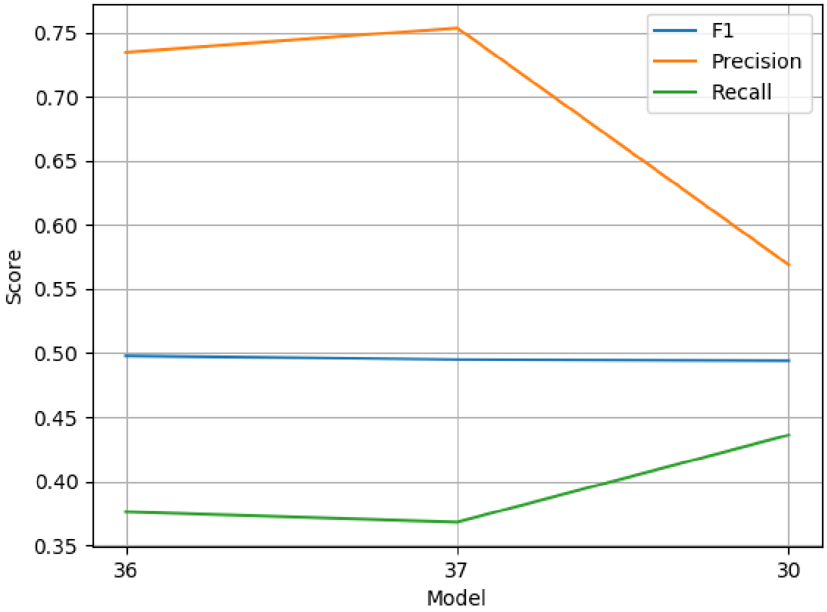


Figure 4.6: Plot of top performances

4.7 Comparison of Models

The top performing models are compared to the current solution in Table 4.7. The current solution of manually inspecting every suspected flaw effectively treats all suspected flaws as true flaws. The amount of true positives thus become identical to the number of true flaws in the testing set, being 63.2% due to class imbalance. Samples of normal rail conditions constitute the remaining 36.8% and are treated as true flaws by the current solution, resulting in false positives. As a result, the current solution have the highest percentage of true positives, but also the most false positives. Model 37 have the highest percentage of false positives and false positives among the top performing models. Model 30 have the highest percentages of true negatives and false negatives.

Table 4.7: Confusion matrix scores for current solution and top performing models

	Current	**Model 36**	**Model 37**	**Model 30**
TP	63.2%	58.4%	59.0%	51.2%
FP	36.8%	22.9%	23.1%	20.7%
TN	0.0%	13.8%	13.5%	16.0%
FN	0.0%	5.0%	4.4%	12.1%

The prediction accuracy for each of the two rail conditions is depicted in Table 4.8. Compared to the current solution, the proposed models all have lower flaw accuracy, with model 37 having the highest. The proposed models all have an accuracy of predicting normal conditions above 35%, while the current solution never consider suspected flaws as normal conditions. A higher flaw accuracy correlate with a lower accuracy of predicting normal conditions, and vice versa.

Table 4.8: Prediction of rail conditions

Condition	**Current**	**Model 36**	**Model 37**	**Model 30**
Flaw	100%	92.1%	93.1%	80.9%
Normal	0%	37.6%	36.9%	43.6%

The utility of the solutions for the three scenarios are illustrated in Figure 4.7. It can be seen that in Scenario 1, where $|u_1(TN)| << |u_1(FN)|$, the current solution has the highest utility. For Scenario 2, where $|u_2(TN)| = |u_2(FN)|$, Model 37 have the highest utility, with Model 36 as a close second. In Scenario 3, where $|u_3(TN)| >> |u_3(FN)|$ the model with highest utility is Model 30.

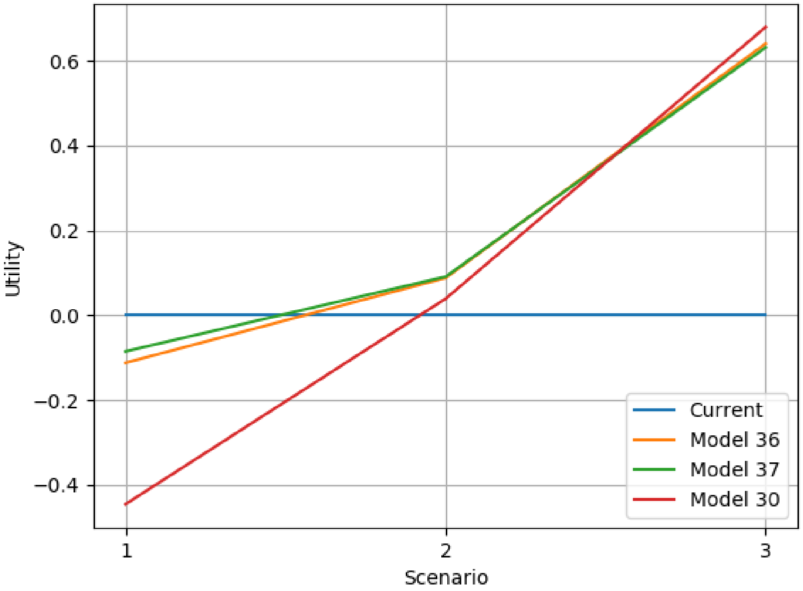


Figure 4.7: Utility values of the different solutions

Chapter 5

Discussion

The objectives of the thesis were successfully completed and the methods described in the case study. The first section of this chapter include the discussion of the research question: *What are the advantages and disadvantages of the proposed model(s)?* The chapter further include the results of different combinations, evaluation of methods used, suggested improvements and further work.

5.1 Models

From Table 4.7, the increase in true positives is seen to be associated with an increase in false positives. The same is true for the prediction accuracy of normal conditions. Logically, if the model has an accuracy below 100%, predicting flaws will result in an increase in both true positives and false positives. The same is true for negatives. As a result, the accuracy of false and positive predictions must be considered. Model 37 has the highest prediction accuracy for both positives and negatives, 71.9% and 75.4%, respectively. The model also has the highest F1 score and will therefore be used as the primary proposed model for this discussion.

The advantage of the proposed models compared to the current solution is the introduction of true negatives, i.e. suspected flaws correctly classified as normal conditions. The true negatives can be disregarded from further inspection, lowering the number of manual inspections needed. The disadvantage of the proposed models is the introduction of false negatives, i.e. actual flaws wrongly classified as normal conditions. As a result, flaws are left uninspected. The highest accuracy for detecting flaws was achieved by model 37 with 93.1% of the flaws predicted to be flaws, as depicted in Table 4.8. As a result, the remaining 6.9% of the flaws are wrongly classified as normal condition, and thus not inspected. The seemingly low accuracy of detecting normal conditions, e.g. 36.9% for model 37, does not have severe implications. By looking at the results in Table 4.7, the remaining 63.1% of the normal conditions are seen to be false positives. The low percentage is therefore not critical as it results in redundant inspections, not uninspected flaws. The accuracy percentage for predicting normal conditions correspond to the percentage of redundant inspections saved. In the case of model 37, the percentage of redundant inspections saved therefore become 36.9%.

The utility value of the different models illustrated in Figure 4.7 show that the choice of model depend on the situation. The current solution can be favorable if minimizing the number of uninspected flaws is significantly more important than minimizing the number of redundant inspections, i.e. safety is highly favoured over cost savings. Model 37 is the appropriate model for when the relationship is equal, while model 30 is appropriate if saving cost is most important. The utility values for true and false negatives are assumed to be dynamic in the real world. The theory state that the rate of flaws are influenced by the loads and temperatures, which are both dynamic. Other factors to consider when establishing the utility values include location, safety concerns and budget. Model 36 is never the optimal choice in this calculation of utility. The model could however be chosen if the training duration needs to be considered. Model 37 has more than 4 times longer training duration than model 36, as depicted in Table 4.6. While the training duration is insignificant with the current number of samples, the duration can be decisive in the future if the number of samples are greatly increased. The threshold for duration should be high, as the model is typically only needed to be re-trained yearly, when new labelled samples become available. However, the shorter duration could be favourable if testing a significant number of new combinations.

5.2 Combinations

The importance of testing different combinations is verified by the large difference in scores. The lowest average F1 score was 0.300, while the highest average F1 score was 0.484. The results thus further confirm the need for trial and error expressed in theory. The average training duration was found to increase with the number of features. The finding is consistent with the theory and the Random forest algorithm constructing trees with a depth correlated with the number of features. The results of different combinations are discussed in this section.

5.2.1 Feature Selection

The performance did not necessarily increase by increasing the number of K best features. An optimal number of features was found and the finding is consistent with the theory stating that an insufficient number of features can lead to underfitting, while an excessive amount can result in overfitting (Suthaharan, 2016a). Selecting K best features was found to outperform resolution reduction for minimizing the number of features. The lowest resolution had the highest performance score. The performance can be explained by the high resolutions introducing redundant features causing overfitting, and the information being preserved despite the lower resolution.

5.2.2 Number of Trees

The Precision score was found to increase with the number of trees. However, as the number of trees were increased sufficiently, the increase in Precision was accompanied by a decrease in Recall. The number of trees generally increased performance, but the level of performance flattened as the number reached 5 000. The correlation can be explained by the properties of the collective random subsets of features chosen during node splitting, when constructing the decision trees, representing the properties of the full feature set when having sufficiently many trees. The drop in performance for 7 500 trees is regarded an outlier and inconsistent with the other results.

5.2.3 Number of Samples

The performance increased with the number of samples. The results are coherent with the theory stating that the model is reliant on a sufficiently large and diverse dataset to achieve generalization (Du and Swamy, 2014). The model was therefore underfitting when trained on a small datasets, but did not experience overfitting from too many samples (Suthaharan, 2016a). Due to the performance increasing with the number of samples and the results not showing a performance decrease indicating overfitting, it is reasonable to assume that increased performance can be achieved by using a larger dataset.

5.2.4 Sensor Subset

The full set of sensors, combining all three methods, proved to have the highest performance scores. The Standard subset had similar scores to the full set, compared to the Squats and VSH subsets which both experienced a drop in Recall and a spike in Precision. The drop in Recall can be explained by the model not predicting as many negatives, which is consistent with the two subsets having significantly lower amount of predictions of negatives, compared to the Standard subset and the full set. The spike in Precision can be explained by the models being overfitted on the most decisive features and accurately predicting only a small number of samples as a result. The splitting into sensor subsets could prove favorable if the aim was to predict the type of flaws.

5.3 Evaluation of Method and Improvements

This section includes the evaluation of methods used to produce the results and suggested improvements.

5.3.1 Subset Selection

Methods for converting the document files to image files could have been explored to possible include them for further consideration. However, as visualized in Figure 3.2, text is present in the B-scans, which had to be removed. The format found in the document files does not, dissimilar to the chosen subset, have a grid that would require removal. Significant pre-processing work was likely saved by choosing a standardized format instead of the other image formats consisting of software screenshots. But as depicted in Figure 3.3, all the other formats with more than 1000 samples are similar to each other, compared to the chosen format. The pre-processing methods used in the case study could thus have been reused for similar formats. The format chosen was also found to have inconsistency in the range of the horizontal axis and the colours. The range varied between 0.5 meters and more than 1.2 kilometers. Furthermore, the format consisted of two B-scans, one for each rail, requiring extracting the right B-scan. Due to inconsistency in which rail is presented in the upper and lower B-scans, the wrong B-scans can have been chosen, exposing the model to wrong data hurting the performance. Additionally, the subset selected was found to have almost twice as high percentage of false positives as the complete dataset provided by Bane NOR. This helped counter imbalance in the dataset, but is not representative of the general situation. The number of false positives was not found to generally outnumber the true positives, which is inconsistent with the theory. Furthermore, the subset used in the case study was relatively small with 1 811 samples. The hybrid system mentioned in the introduction used 38 000 samples (Jarmulak et al., 1998). The B-scans could also be sub-optimal due to error from the inspection companies.

The flaw property in the inspection data used as sample labels was chosen due to being the only property indicating whether the manual inspection was redundant or not. Flaws below the threshold for registration could have been included as false positives as they cause redundant inspections. They were included as true positives due to representing flaws, although small. Alternatively, all three categories could have been used as classes, changing the classification problem from binary to 3 classifications. The labels of the data could be wrong if there is error in the inspection data, harming the learning of the algorithm.

5.3.2 Pre-Processing Methods

Finding and extracting the vertical range used in the B-scans is redundant if selecting the K best features later. However, it reduced the computational complexity encountered when training on different resolutions and sensor subsets. The removal of the lower part of the pre-processing method should not be included if performed again in the future, because the need for you will have to check the maximum horizontal value used for each sample added to the dataset and change the algorithm if it is changed. If changed, the whole dataset needs to be pre-processed again. It was also proven better to use K best features anyway.

Oversampling or undersampling methods were not performed on the dataset to reduce the class imbalance. Oversampling, i.e. creating more samples of the minority class, would not be possible due to the complexity of producing artificial B-scans and the expert knowledge needed. Undersampling methods, i.e. removing samples of the majority class, were discarded due to the small number of samples in the dataset.

Other algorithms could have been used for resolution reduction, feature selection and training of the models. More complex algorithms could replace the nearest neighbour algorithm used for resolution reduction and the ANOVA F-score algorithm used for feature selection. The introduction of more complex algorithms would however increase the computational complexity and thus be more resource demanding. The Random forest algorithm is found to be generally good at training models, but other algorithms could have performed better on the specific case.

5.3.3 Training and Evaluation of Models

The steps in the process used to develop the models, illustrated in Figure 2.20, are common for machine learning. However, being an iterative process, the sequence of the steps and iterations could have been performed otherwise. The training sequence for finding optimal values could have been done in any sequence. Selecting the K best features in the first training sequence and lowering the resolution in the second sequence were chosen because pre-processing is typically performed first, because it reduces the computational complexity and typically has a high impact on performance. Furthermore, the ratio between training set and testing set was not altered throughout the study. The ratio could have been adjusted, but would introduce another variable to the possible combinations and the 80:20 ratio was suggested in literature. In addition, more than 3 models could have been used to produce the averages of each model used for comparison. The evaluation scores were revised to fit the problem better. The original performance measures could have been used, but would have given a high score for sub-optimal models, based on the aim of the thesis.

5.3.4 Further Work

The proposed models could be integrated with other models and solutions, or be employed independently. A possible fit for integration is the model proposed in Vatn et al. (2006) for determining the optimal strategy for ultrasound inspection of rails based on cost and risk estimation. The optimization model could be used to determine which rail flaw detection model to use. The testing of other subsets, combinations and algorithms is requested for further work to potentially increase the performance. Furthermore, 65 of the 66 properties in the inspection data are not used in this case study and could be explored.

The utility values are needed to decide on the optimal model. It would also be beneficial to know if funds saved could be spent on other, perhaps more effective, safety initiatives. This thesis can be used as a guideline if Bane NOR would like to start other machine learning projects. The proposed models could also inspire the inspection companies to improve their models. The optimal solution would be if the inspection companies were accurate enough to fully rely on, thus eliminating the need for manual inspections.

Chapter 6

Conclusion

Models were successfully developed using machine learning to accurately detect flaws in railway rails from B-scans. The top performing models showed a significant reduction in false positives, while maintaining high prediction accuracy of flaws. The number of redundant inspections can thus be reduced without a large decline in safety. The choice of model depend on the utility relation between flaws left uninspected and redundant manual inspections saved. The appropriate model is presented for three different scenarios. As rails are predicted to experience heavier loads and speeds in the future, the number of flaws is set to increase and the need for accurate prediction models grows. The thesis can function as a guideline for railway companies or inspection companies with a need to lower false positives and thus increase the efficiency of inspections.

A wide range of combinations of subsets and parameters were tested to achieve the highest performance scores. The process of applying machine learning to the real-world problem is found to be problem-specific and characterized by trial and error, consistent with theory and conclusions in literature. The Random forest algorithm was found to perform satisfactory on this dataset. One parameter of the algorithm, the number of trees, was successfully tuned to increase performance.

Different pre-processing methods were tested. Selecting the K best features was found to outperform reducing the resolution. Increased number of samples and decreased number of features were found to produce good results. The need for pre-processing could have been greatly reduced, and the results likely improved, if the inspection companies would deliver the B-scans in a more suitable format with consistent vertical ranges and colours. Improvements and further work has been suggested and is requested.

Bibliography

Alpaydin, E. (2009). *Introduction to machine learning*. MIT press.

Alpaydin, E. (2014). Introduction to machine learning (3rd edition).

Archer, K. J. and Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics Data Analysis*, 52(4):2249 – 2260.

Bane NOR (2017). 2017 Brochure Bane NOR - We create the railway of the future. accessed 21.03.2019.

Bane NOR (2018). Om Bane NOR. `https://www.banenor.no/Om-oss/Om_Bane-NOR/`, accessed 20.03.2019.

Baptista, R., Santos, T., Marques, J., Guedes, M., and Infante, V. (2018). Fatigue behavior and microstructural characterization of a high strength steel for welded railway rails. *International Journal of Fatigue*, 117:1 – 8.

Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29.

BBC (2017). Granville: The rail disaster that changed Australia. `https://www.bbc.com/news/world-australia-38645976`, accessed 11.03.2019.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122.

CANNON, D. F., EDEL, K.-O., GRASSIE, S. L., and SAWLEY, K. (2003). Rail defects: an overview. *Fatigue & Fracture of Engineering Materials & Structures*, 26(10):865–886.

Caruana, R. and Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 161–168, New York, NY, USA. ACM.

Chen, Y., Ma, H.-W., and Zhang, G.-M. (2014). A support vector machine approach for classification of welding defects from ultrasonic signals. *Nondestructive Testing and Evaluation*, 29(3):243–254.

Chou, J.-H., Ghaboussi, J., and Clark, R. (1999). *Application of Neural Networks to the Inspection of Railroad Rail*, pages 2121–2128. Springer US, Boston, MA.

Clark, R. (2004). Rail flaw detection: overview and needs for future developments. *NDT E International*, 37(2):111 – 118.

Clark, R. and Singh, S. (2003). The inspection of thermite welds in railroad rail a perennial problem. *Insight - Non-Destructive Testing and Condition Monitoring*, 45(6):387–393.

Coutanche, M. N., Thompson-Schill, S. L., and Schultz, R. T. (2011). Multi-voxel pattern analysis of fmri data predicts clinical symptom severity. *NeuroImage*, 57(1):113–123.

Cygan, H., Girardi, L., Aknin, P., and Simard, P. (2003). B-scan ultrasonic image analysis for internal rail defect detection. In *B-scan ultrasonic image analysis for internal rail defect detection*.

Dayal, R. K. and Parvathavarthini, N. (2003). Hydrogen embrittlement in power plant steels. *Sadhana*, 28(3):431–451.

Du, K.-L. and Swamy, M. N. S. (2014). *Fundamentals of Machine Learning*, pages 15–65. Springer London, London.

Faiz, R. B. and Singh, S. (2009). Predictive maintenance management of rail profile in uk rail. In *2009 International Conference on Computing, Engineering and Information*, pages 370–375.

Giben, X., Patel, V. M., and Chellappa, R. (2015). Material classification and semantic segmentation of railway track images with deep convolutional neural networks. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 621–625.

Gordon, G., Canumalla, S., and Tittmann, B. (1993). Ultrasonic c-scan imaging for material characterization. *Ultrasonics*, 31(5):373 – 380. Special Issue Acousto Optics/Imaging.

Greisen, C., Lu, S., Duan, H., Farritor, S., Arnold, R., GeMeiner, B., Clark, D., Toth, T., Hicks, K., Sussmann, T., Fateh, M., and Carr, G. (2009). Estimation of rail bending stress from real-time vertical track deflection measurement.

Hajizadeh, S., NÃºÃ±ez, A., and Tax, D. M. (2016). Semi-supervised rail defect detection from imbalanced image data. *IFAC-PapersOnLine*, 49(3):78 – 83. 14th IFAC Symposium on Control in Transportation SystemsCTS 2016.

Hall, K. (1976). Crack depth measurement in rail steel by rayleigh waves aided by photoelastic visualization. *Non-Destructive Testing*, 9(3):121 – 126.

Holmgren, M. (2005). Maintenanceâ€related losses at the swedish rail. *Journal of Quality in Maintenance Engineering*, 11(1):5–18.

Hong, X., Xiao, G., Haoyu, W., Xing, L., and Sixing, W. (2018). Fatigue damage analysis and life prediction of e-clip in railway fasteners based on abaqus and fe-safe. *Advances in Mechanical Engineering*, 10(3):1687814018767249.

Jarmulak, J., J. H. Kerckhoffs, E., and Paul van't Veen, P. (1998). Hybrid knowledge based system for automatic classificaton of b-scan images from ultrasonic rail inspection. pages 1121–1126.

Johnson, K. J. and Synovec, R. E. (2002). Pattern recognition of jet fuels: comprehensive gcÃ—gc with anova-based feature selection and principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 60(1):225 – 237. Fourth International Conference on Environ metrics and Chemometrics held in Las Vegas, NV, USA, 18-20 September 2000.

Kaewunruen, S., Gamage, E. K., and Remennikov, A. M. (2016). Structural behaviours of railway prestressed concrete sleepers (crossties) with hole and web openings. *Procedia Engineering*, 161:1247 – 1253. World Multidisciplinary Civil Engineering-Architecture-Urban Planning Symposium 2016, WMCAUS 2016.

KING, R. D., FENG, C., and SUTHERLAND, A. (1995). Statlog: Comparison of classification algorithms on large real-world problems. *Applied Artificial Intelligence*, 9(3):289–333.

Kitahara, M., Achenbach, J. D., Guo, Q. C., Peterson, M. L., Notake, M., and Takadoya, M. (1992). *Neural Network for Crack-Depth Determination from Ultrasonic Backscattering Data*, pages 701–708. Springer US, Boston, MA.

Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24.

Krautkrämer, J. and Krautkrämer, H. (2013). *Ultrasonic testing of materials*. Springer Science & Business Media.

Kubat, M. (2017). An introduction to machine learning.

Lehmann, C., Koenig, T., Jelic, V., Prichep, L., John, R. E., Wahlund, L.-O., Dodge, Y., and Dierks, T. (2007). Application and comparison of classification algorithms for recognition of alzheimer's disease in electrical brain activity (eeg). *Journal of Neuroscience Methods*, 161(2):342 – 350.

Lidén, T. (2015). Railway infrastructure maintenance - a survey of planning problems and conducted research. *Transportation Research Procedia*, 10:574 – 583. 18th Euro Working Group on Transportation, EWGT 2015, 14-16 July 2015, Delft, The Netherlands.

Lim, T.-S., Loh, W.-Y., and Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–228.

Lin Jie, Luo Siwei, Li Qingyong, Zhang Hanqing, and Ren Shengwei (2009). Real-time rail head surface defect detection: A geometrical approach. In *2009 IEEE International Symposium on Industrial Electronics*, pages 769–774.

Lison, P. (2015). An introduction to machine learning.

M Mitchell, T. (2006). The discipline of machine learning.

Macha, E., Bedkowski, W., and Lagoda, T. (1999). *Multiaxial Fatigue and Fracture*. European Structural Integrity Society. Elsevier Science.

Maruyama, T., Kurita, T., Kozaki, S., Andou, K., Farjami, S., and Kubo, H. (2008). Innovation in producing crane rail fishplate using feâ€“mnâ€“siâ€“cr based shape memory alloy. *Materials Science and Technology*, 24(8):908–912.

Mwangi, B., Tian, T. S., and Soares, J. C. (2014). A review of feature reduction techniques in neuroimaging. *Neuroinformatics*, 12(2):229–244.

Orringer, O., Morris, J., and Steele, R. (1984). Applied research on rail fatigue and fracture in the united states. *Theoretical and Applied Fracture Mechanics*, 1(1):23 – 49.

Papaelias, M. P., Roberts, C., and Davis, C. L. (2008). A review on non-destructive evaluation of rails: State-of-the-art and future development. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 222(4):367–384.

Rodriguez-Galiano, V., Ghimire, B., Rogan, J., Chica-Olmo, M., and Rigol-Sanchez, J. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 67:93 – 104.

Rokach, L. and Maimon, O. Z. (2008). *Data mining with decision trees: theory and applications*, volume 69. World scientific.

Singh, G. P. and Manning, R. C. (1983). An artificial intelligence approach to ultrasonic weld evaluation.

Smola, A. and Vishwanathan, S. (2008). Introduction to machine learning. *Cambridge University, UK*, 32:34.

Sokolova, M. and Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437.

Steenbergen, M. (2016). Rolling contact fatigue in relation to rail grinding. *Wear*, 356-357:110 – 121.

Suthaharan, S. (2016a). *Modeling and Algorithms*, pages 123–143. Springer US, Boston, MA.

Suthaharan, S. (2016b). *Supervised Learning Algorithms*, pages 183–206. Springer US, Boston, MA.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Takadoya, M., Yabe, Y., Kitahara, M., Achenbach, J. D., Guo, Q. C., and Peterson, M. L. (1995). *An Artificial Intelligence Technique to Characterize Surface-Breaking Cracks*, pages 771–778. Springer US, Boston, MA.

Toliyat, H. A., Abbaszadeh, K., Rahimian, M. M., and Olson, L. E. (2003). Rail defect diagnosis using wavelet packet decomposition. *IEEE Transactions on Industry Applications*, 39(5):1454–1461.

Tutumluer, E., Huang, H., Hashash, Y., and Ghaboussi, J. (2006). Aggregate shape effects on ballast tamping and railroad track lateral stability. In *AREMA Annual Conference, Loisville, KY, Sept*, pages 17–20.

Wang, L., Pyzalla, A., Stadlbauer, W., and Werner, E. (2003). Microstructure features on rolling surfaces of railway rails subjected to heavy loading. *Materials Science and Engineering: A*, 359(1):31 – 43.

Yu, L. and Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224.

Zerbst, U., Lundén, R., Edel, K.-O., and Smith, R. A. (2009a). Introduction to the damage tolerance behaviour of railway rails–a review. *Engineering fracture mechanics*, 76(17):2563–2601.

Zerbst, U., LundÃ©n, R., Edel, K.-O., and Smith, R. (2009b). Introduction to the damage tolerance behaviour of railway rails â€“ a review. *Engineering Fracture Mechanics*, 76(17):2563 – 2601. Special Issue on the Damage Tolerance of Railway Rails.

Zgonc, K., Achenbach, J. D., and Lee, Y.-C. (1995). *Crack Sizing Using a Neural Network Classifier Trained with Data Obtained from Finite Element Models*, pages 779–786. Springer US, Boston, MA.

Appendix A

Acronyms

RCF Rolling contact fatigue

VSH Vertical split head

NDE Non-destructive evaluation

PEC Pulsed eddy current

TP True positive

FP False positive

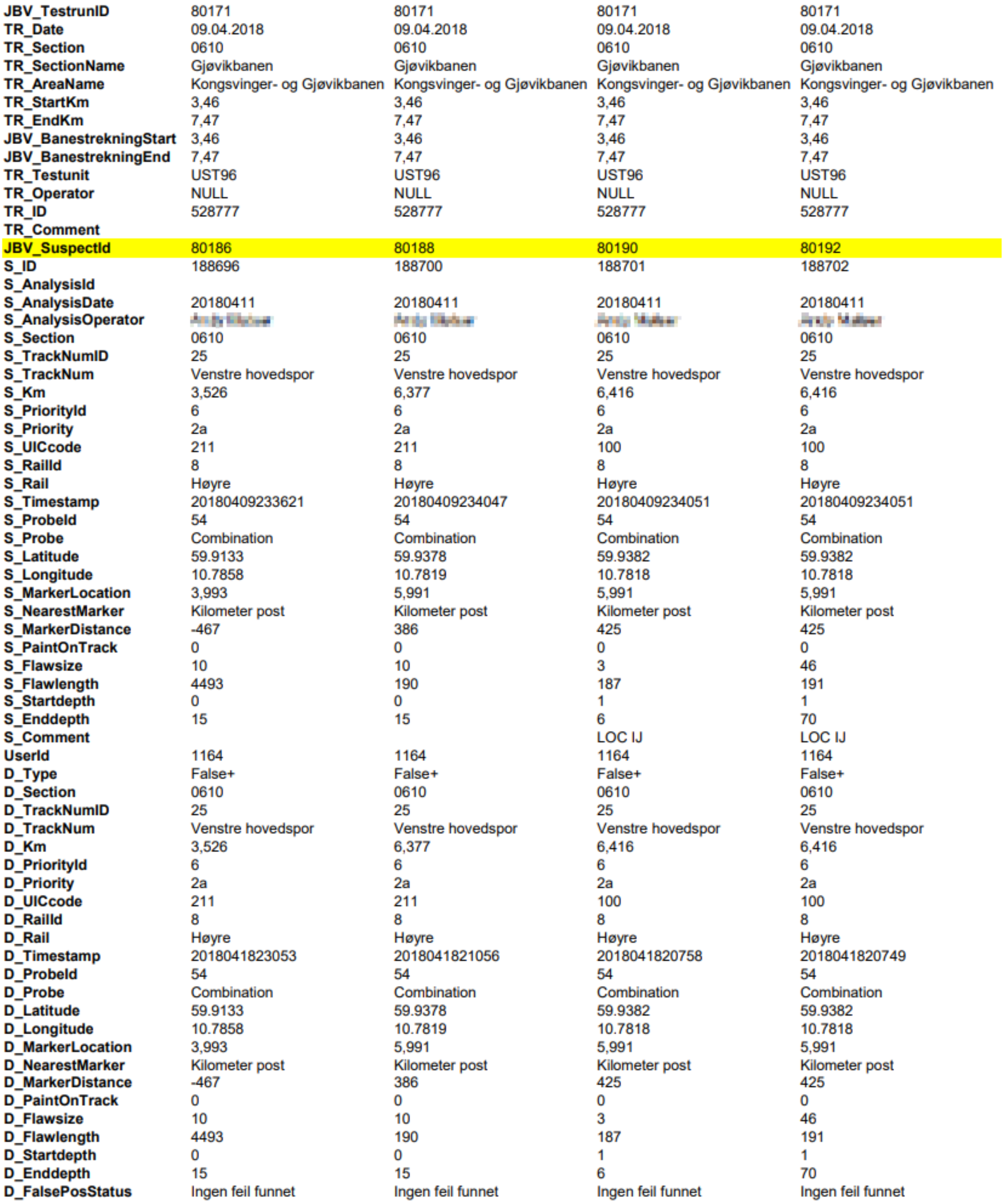
TN True negative

FN False negative

RGB Red, green, blue

Appendix B

Inspection Data



JBV_TestrunID	80171	80171	80171	80171
TR_Date	09.04.2018	09.04.2018	09.04.2018	09.04.2018
TR_Section	0610	0610	0610	0610
TR_SectionName	Gjøvikbanen	Gjøvikbanen	Gjøvikbanen	Gjøvikbanen
TR_AreaName	Kongsvinger- og Gjøvikbanen	Kongsvinger- og Gjøvikbanen	Kongsvinger- og Gjøvikbanen	Kongsvinger- og Gjøvikbanen
TR_StartKm	3,46	3,46	3,46	3,46
TR_EndKm	7,47	7,47	7,47	7,47
JBV_BanestrekningStart	3,46	3,46	3,46	3,46
JBV_BanestrekningEnd	7,47	7,47	7,47	7,47
TR_Testunit	UST96	UST96	UST96	UST96
TR_Operator	NULL	NULL	NULL	NULL
TR_ID	528777	528777	528777	528777
TR_Comment				
JBV_SuspectId	80186	80188	80190	80192
S_ID	188696	188700	188701	188702
S_AnalysisId				
S_AnalysisDate	20180411	20180411	20180411	20180411
S_AnalysisOperator	[redacted]	[redacted]	[redacted]	[redacted]
S_Section	0610	0610	0610	0610
S_TrackNumID	25	25	25	25
S_TrackNum	Venstre hovedspor	Venstre hovedspor	Venstre hovedspor	Venstre hovedspor
S_Km	3,526	6,377	6,416	6,416
S_PriorityId	6	6	6	6
S_Priority	2a	2a	2a	2a
S_UICcode	211	211	100	100
S_RailId	8	8	8	8
S_Rail	Høyre	Høyre	Høyre	Høyre
S_Timestamp	20180409233621	20180409234047	20180409234051	20180409234051
S_ProbeId	54	54	54	54
S_Probe	Combination	Combination	Combination	Combination
S_Latitude	59.9133	59.9378	59.9382	59.9382
S_Longitude	10.7858	10.7819	10.7818	10.7818
S_MarkerLocation	3,993	5,991	5,991	5,991
S_NearestMarker	Kilometer post	Kilometer post	Kilometer post	Kilometer post
S_MarkerDistance	-467	386	425	425
S_PaintOnTrack	0	0	0	0
S_Flawsize	10	10	3	46
S_Flawlength	4493	190	187	191
S_Startdepth	0	0	1	1
S_Enddepth	15	15	6	70
S_Comment			LOC IJ	LOC IJ
UserId	1164	1164	1164	1164
D_Type	False+	False+	False+	False+
D_Section	0610	0610	0610	0610
D_TrackNumID	25	25	25	25
D_TrackNum	Venstre hovedspor	Venstre hovedspor	Venstre hovedspor	Venstre hovedspor
D_Km	3,526	6,377	6,416	6,416
D_PriorityId	6	6	6	6
D_Priority	2a	2a	2a	2a
D_UICcode	211	211	100	100
D_RailId	8	8	8	8
D_Rail	Høyre	Høyre	Høyre	Høyre
D_Timestamp	2018041823053	2018041821056	2018041820758	2018041820749
D_ProbeId	54	54	54	54
D_Probe	Combination	Combination	Combination	Combination
D_Latitude	59.9133	59.9378	59.9382	59.9382
D_Longitude	10.7858	10.7819	10.7818	10.7818
D_MarkerLocation	3,993	5,991	5,991	5,991
D_NearestMarker	Kilometer post	Kilometer post	Kilometer post	Kilometer post
D_MarkerDistance	-467	386	425	425
D_PaintOnTrack	0	0	0	0
D_Flawsize	10	10	3	46
D_Flawlength	4493	190	187	191
D_Startdepth	0	0	1	1
D_Enddepth	15	15	6	70
D_FalsePosStatus	Ingen feil funnet	Ingen feil funnet	Ingen feil funnet	Ingen feil funnet

Figure B.1: Examples of inspection data

Appendix C

Individual Model Performance

C.1 K Best Features

Table C.1: Performance of different K best features

Model	K	TP	FP	TN	FN	Precision	Recall	F1	Duration
1	1 000	0.548	0.256	0.110	0.085	0.563	0.301	0.392	0.426
2	1 000	0.529	0.275	0.091	0.105	0.465	0.248	0.324	0.347
3	1 000	0.540	0.270	0.096	0.094	0.507	0.263	0.347	0.346
4	10 000	0.565	0.242	0.124	0.069	0.643	0.338	0.443	1.856
5	10 000	0.554	0.237	0.129	0.080	0.618	0.353	0.450	1.548
6	10 000	0.570	0.234	0.132	0.06	0.676	0.361	0.471	1.652
7	13 500	0.565	0.234	0.132	0.069	0.658	0.361	0.466	3.529
8	13 500	0.570	0.234	0.132	0.063	0.676	0.361	0.471	3.681
9	13 500	0.565	0.240	0.127	0.069	0.648	0.346	0.451	3.413
10	15 500	0.576	0.240	0.127	0.058	0.687	0.346	0.460	3.588
11	15 500	0.579	0.234	0.132	0.055	0.706	0.361	0.478	3.530
12	15 500	0.570	0.234	0.132	0.063	0.676	0.361	0.471	3.446
13	17 500	0.562	0.234	0.132	0.072	0.649	0.361	0.464	2.450
14	17 500	0.562	0.234	0.132	0.072	0.649	0.361	0.464	2.498
15	17 500	0.565	0.240	0.127	0.069	0.648	0.346	0.451	2.383
16	25 000	0.573	0.237	0.129	0.061	0.681	0.353	0.465	3.246
17	25 000	0.551	0.234	0.132	0.083	0.615	0.361	0.455	3.629
18	25 000	0.559	0.245	0.121	0.074	0.620	0.331	0.431	3.084
19	50 000	0.548	0.229	0.138	0.085	0.617	0.376	0.467	5.115
20	50 000	0.565	0.237	0.129	0.069	0.653	0.353	0.459	5.020
21	50 000	0.556	0.234	0.132	0.077	0.632	0.361	0.459	4.957
22	75 000	0.548	0.242	0.124	0.085	0.592	0.338	0.431	6.687
23	75 000	0.556	0.242	0.124	0.077	0.616	0.338	0.437	6.707
24	75 000	0.562	0.251	0.116	0.072	0.618	0.316	0.418	6.579
25	100 000	0.548	0.245	0.121	0.085	0.587	0.331	0.423	8.120
26	100 000	0.554	0.262	0.105	0.080	0.567	0.286	0.380	8.061
27	100 000	0.559	0.251	0.116	0.074	0.609	0.316	0.416	8.262

C.2 Number of Trees

Table C.2: Performance of different number of trees

Model	Trees	TP	FP	TN	FN	Precision	Recall	F1	Duration
28	10	0.521	0.234	0.132	0.113	0.539	0.361	0.432	0.762
29	10	0.512	0.223	0.143	0.121	0.542	0.391	0.454	0.657
30	10	0.512	0.207	0.160	0.121	0.569	0.436	0.494	0.651
31	100	0.576	0.240	0.127	0.058	0.687	0.346	0.460	3.588
32	100	0.579	0.234	0.132	0.055	0.706	0.361	0.478	3.530
33	100	0.570	0.234	0.132	0.063	0.676	0.361	0.471	3.446
34	1 000	0.584	0.242	0.124	0.050	0.714	0.338	0.459	32.821
35	1 000	0.581	0.234	0.132	0.052	0.716	0.361	0.480	38.752
36	1 000	0.584	0.229	0.138	0.050	0.735	0.376	0.498	38.744
37	5 000	0.590	0.231	0.135	0.044	0.754	0.368	0.495	159.590
38	5 000	0.584	0.237	0.129	0.050	0.723	0.353	0.475	161.844
39	5 000	0.584	0.237	0.129	0.050	0.723	0.353	0.475	171.152
40	7 500	0.590	0.240	0.127	0.044	0.742	0.346	0.472	262.797
41	7 500	0.584	0.240	0.127	0.050	0.719	0.346	0.467	278.943
42	7 500	0.584	0.234	0.132	0.050	0.727	0.361	0.482	282.131
43	10 000	0.590	0.234	0.132	0.044	0.750	0.361	0.487	375.299
44	10 000	0.587	0.234	0.132	0.047	0.738	0.361	0.485	430.353
45	10 000	0.581	0.234	0.132	0.052	0.716	0.361	0.480	394.288
46	15 000	0.587	0.237	0.129	0.047	0.734	0.353	0.477	491.114
47	15 000	0.587	0.234	0.132	0.047	0.738	0.361	0.485	497.506
48	15 000	0.590	0.237	0.129	0.044	0.746	0.353	0.480	529.541

C.3 Number of Samples

Table C.3: Performance of different number of samples

Model	Samples	TP	FP	TN	FN	Precision	Recall	F1	Duration
49	226	0.622	0.256	0.067	0.056	0.545	0.207	0.300	27.415
50	226	0.622	0.256	0.067	0.056	0.545	0.207	0.300	27.552
51	226	0.622	0.256	0.067	0.056	0.545	0.207	0.300	27.380
52	453	0.594	0.272	0.083	0.050	0.625	0.234	0.341	81.815
53	453	0.600	0.278	0.078	0.044	0.636	0.219	0.326	79.104
54	453	0.600	0.278	0.078	0.044	0.636	0.219	0.326	74.995
55	906	0.563	0.263	0.115	0.059	0.660	0.304	0.416	132.787
56	906	0.563	0.263	0.115	0.059	0.660	0.304	0.416	142.997
57	906	0.563	0.263	0.115	0.059	0.660	0.304	0.416	129.771
58	1811	0.590	0.234	0.132	0.044	0.750	0.361	0.487	375.299
59	1811	0.587	0.234	0.132	0.047	0.738	0.361	0.485	430.353
60	1811	0.581	0.234	0.132	0.052	0.716	0.361	0.480	394.288

C.4 Resolution

Table C.4: Performance of different resolutions

Model	Resolution	TP	FP	TN	FN	Precision	Recall	F1	Duration
61	(180, 50)	0.488	0.218	0.149	0.146	0.505	0.406	0.450	6.744
62	(180, 50)	0.504	0.215	0.152	0.129	0.541	0.414	0.469	6.944
63	(180, 50)	0.501	0.231	0.135	0.132	0.506	0.369	0.427	6.610
64	(360, 101)	0.507	0.229	0.138	0.127	0.521	0.376	0.437	15.407
65	(360, 101)	0.504	0.242	0.124	0.129	0.490	0.339	0.401	16.154
66	(360, 101)	0.512	0.248	0.118	0.121	0.494	0.322	0.390	15.665
67	(719, 202)	0.515	0.240	0.127	0.118	0.518	0.346	0.415	38.424
68	(719, 202)	0.504	0.240	0.127	0.129	0.496	0.346	0.408	39.867
69	(719, 202)	0.499	0.240	0.127	0.135	0.494	0.346	0.407	39.079
70	(1437, 404)	0.485	0.229	0.138	0.149	0.481	0.376	0.422	125.058
71	(1437, 404)	0.507	0.253	0.113	0.127	0.471	0.309	0.373	115.490
72	(1437, 404)	0.507	0.240	0.127	0.127	0.500	0.346	0.409	121.269

C.5 Sensor Subset

Table C.5: Performance of different sensor subsets

Model	Sensor subset	TP	FP	TN	FN	Precision	Recall	F1	Duration
73	Standard	0.479	0.220	0.146	0.154	0.486	0.398	0.438	3.852
74	Standard	0.485	0.226	0.140	0.149	0.486	0.383	0.429	3.980
75	Standard	0.482	0.218	0.149	0.152	0.495	0.406	0.446	4.450
76	Squats	0.556	0.289	0.077	0.077	0.500	0.211	0.296	3.060
77	Squats	0.534	0.275	0.091	0.099	0.478	0.248	0.327	2.975
78	Squats	0.562	0.267	0.099	0.072	0.581	0.271	0.369	3.071
79	VSH	0.579	0.284	0.083	0.055	0.600	0.226	0.328	3.199
80	VSH	0.567	0.281	0.085	0.066	0.564	0.233	0.330	3.180
81	VSH	0.579	0.278	0.088	0.055	0.615	0.241	0.346	3.156