MASTER'S THESIS

Incident End Time Prediction During the Incident Recovery Process

T. Kraai (Tim)

February 2021

Supervisors University of Twente

Dr. E. Topan

Dr. C.G.M. Groothuis-Oudshoorn

UNIVERSITY OF TWENTE.

Supervisors ProRail

Saskia Wevers

Martijn van der Weide

ProRail

Management Summary

Railway incidents can have a big impact on train operations. By communicating prognoses of the incident end time, ProRail informs railway operators and travellers about the expected end time of an incident. For each incident type, a decision tree is developed by the "Consultants in Qualitative Methods" company (CQM) which determines an initial prognosis during the intake of an incident. At the beginning of an incident, estimating the end time of an incident is difficult because limited information is available. For a prognosis to be reliable, it must be given *in time* (35 minutes before the end of the incident) and it has to be *precise* (the prognosed end time is later than the actual end time). A reliable prognosis of the end time of an incident makes it possible to create an overlap between finishing the recovery activities and planning the restart of the trains. Unreliable prognoses of the incident end time lead to delay. Because of the importance of reliable prognoses and the complexity of the incident recovery process, prognoses during the incidents are currently given by an incident coordinator based on expertise. A reliable prediction of the incident end time determined with a data-based model, can support the coordinator with giving *in time* and *precise* prognoses. Therefore, the goal of this project is defined as:

> *Create a data-based model based on literature and previous research at ProRail that provides predictions of the incident end time to support in time and precise prognoses.*

Method

Data analysis of the current reliability of the prognoses shows that most prognoses are *precise* but are not given *in time*. This project focusses on the incident type collision with a person because of the impact that this incident type has on the train operations and the amount of data that is available during this type of incident. The data of these incidents is pre-processed by removing outliers and imputing missing data. The review of the literature on prediction models and previous research at ProRail show the use of machine learning methods for the development of a data-based incident end time prediction model. Based on this, several machine learning methods are selected. The prediction performance of these methods is determined with cross-validation on all the data of incidents from June 2017 to November 2020. The prediction performance on all data showed the highest performance for the eXtreme Gradient Boosting (XGBoost) method. This method is used to develop a model from which feature importance on all data is analysed. With the model, predictions are determined during chronological stages of the incident. For each stage, feature importance is analysed to identify features with a big impact on the predictions and 90% prediction intervals are generated to communicate the uncertainty of the predictions. The XGBoost model is also used to predict the incident end time when new information about the incident becomes known for three new incidents in December 2020. Lastly, prognoses overpredict the actual duration of an incident to be *precise*. The model however predicts the incident end time by penalizing under- and overprediction evenly. Thus, shifting of the distribution of the residuals is proposed to achieve the desired percentage of overprediction.

Results

The feature importance on all data shows that the updated and final prognoses have a high impact on the prediction accuracy of the developed model. Although the prognoses are important, a clear improvement in predictions and a decrease in the width of the prediction intervals can be seen at stage 2, when the first people arrive at the incident location. After this stage, the predictions slowly improve when more information becomes available.

The features that show the highest impact on the prediction performance of the model at stage 2 are the number of deceased victims and the degree of fragmentation of the body of the victim(s). Important features at other stages are the estimated arrival time and the prognosed finish time of the mortician and the actual arrival time of the AL.

At the final stage, when all data about the incidents is included, a clear split can be seen in the incidents with wide prediction intervals between incidents with a duration < 100 minutes and incidents > 200 minutes. The number of deceased victims is found to be the main feature to explain this split in duration. During the new incidents, the predictions of the incident end time converges to the actual incident end time. The biggest improvement in prediction for these incidents appears when the degree of fragmentation and the prognoses become available.

The predictions from the XGBoost model are given *in time* because this model predicts the incident end time at every moment during the incident. The incident end time is, however, predicted with even under and overprediction while prognoses focus on overprediction. Therefore, the predictions from the XGBoost model cannot directly be compared with the prognoses of the AL. However, a desired percentage of overprediction can be achieved by shifting the distribution of the residuals of the predictions to obtain *precise* prognoses.

Recommendations

This project shows that an XGBoost model can be used for collision with a person incidents to support *in time* prognoses. Further research is recommended to determine if the XGBoost model can also support *in time* prognoses for other incident types.

For the incident type collision with a person, the degree of fragmentation shows a clear improvement in the prediction performance. The CQM decision tree for collision with a person includes the feature degree of fragmentation. However, during the intake, this feature is not known and therefore this CQM decision tree is currently not implemented. It is recommended to ProRail to implement this CQM decision tree at the moment the degree of fragmentation becomes available. Other features that could be used to extend this model are the arrival time of the AL, the estimated time of arrival of the mortician and the prognosed end time of the work of the mortician.

The updated and final prognoses show to be important features for prediction. Therefore, it is recommended to ProRail to keep these prognoses. Further analysis of predictions from the XGBoost model during new incidents can provide more insight into important moments for which a data-based model can support the AL with extra prognoses updates.

Lastly, overprediction of the actual duration of the incident is desired to prevent delay. However, high overprediction leads to additional waiting time before trains can be restarted. Therefore, it is recommended to ProRail to specify the penalty for different levels of under- and overprediction. With these levels a custom objective function can be defined, with which the XGBoost model can determine a *precise* prognosis directly.

Preface

It is a pleasure to present my master's thesis to you. This thesis is the result of half a year of research at ProRail and marks the end of my student time at the University of Twente. I am very thankful for the opportunity to perform my final internship at ProRail. During my internship, I have spoken with many approachable people that were all very open to help and share their knowledge via video calls. This made me feel involved in the process while working from home.

First, I would like to thank my daily supervisors at ProRail, Saskia Wevers and Martijn van der Weide. They motivated me to explore the incident process and the company by inviting me to meetings of other projects and showed great interest in the findings of my research. Secondly, I would like to thank my supervisors of the university, Engin Topan and Karin Groothuis-Oudshoorn for their advice, comments and feedback on my thesis. As sparring partners, they helped me to improve my understanding of the used methods.

Finally, I am very grateful for the support of my family, who challenged me and helped with improving the quality of my thesis, as well as my friends and girlfriend for the mental support and the interest they showed in the topic.

Tim Kraai
Utrecht, February 2021

List of Figures

List of Tables

Abbreviations

In this thesis, many abbreviations are used. Because ProRail is a Dutch company, an English translation is provided. But to prevent confusions, the Dutch abbreviations and Dutch name will be given in brackets. The list shows an overview of the abbreviations and gives their English translations.

AL	Algemeen Leider (General Leader)
DT	Decision Tree
GB	Gradient Boosting
Lasso	Lasso Regression
MKS	Meldkamer Spoor (Railway Alarm Room)
ANN	Artificial Neural Network
PI	Prediction Interval
RF	Random Forest
RFE	Recursive Feature Elimination
SVM	Support Vector Machine
TIS	Trein Incident Scenario (Train Incident Scenario)
TOBS	Ten Onrechte Bezet Spoor (Train Vacancy Detection Failures)
TRDL	Treindienstleider (Railway Traffic Controller)
XGB	eXtreme Gradient Boosting

Table of Contents

1 Introduction

In this chapter, the role of ProRail in the railway system of the Netherlands is described, followed by the motivation for this project in Section 1.2. In Section 1.3, the problem statement and goal of this project are formulated. Section 1.4 describes the methodology for this project and the structure. In Section 1.5, the approach for solving the problem and research questions are defined. Lastly, the scope of this project is described in Section 1.6.

At ProRail many abbreviations are used. Because ProRail is a Dutch company, an English translation is provided. However, to prevent confusions, the Dutch abbreviation and Dutch name will be given in brackets. For example, General Leader (AL, Algemeen Leider). After the explanation, the Dutch abbreviations will be used throughout the report. A full list of all the abbreviations used can be found on page v.

1.1 About ProRail

The Netherlands has one of the world's busiest railway networks. Every day 1 million people travel by train and 100.000 tons of goods are transported over the 7000 kilometres of track (ProRail, 2019a).

In the next 20 years, the population of the Netherlands is expected to increase by 1.6 million. This increase is mostly expected in the Randstad, the urban area in the West of the Netherlands (Kooiman et al., 2016). For many of these people, the train will be a vital mode of transport to commute and travel (van Ammelrooy, 2020). This will lead to an increase of 25-40% of passengers. Besides the number of passengers, the transport of goods is expected to be doubled in 2040. These goods mostly originate from the port of Rotterdam and have to pass through the crowded Randstad to Germany and the rest of Europe (ProRail, 2019a).

ProRail B.V. is a private company, with the Dutch government as the only shareholder that facilitates the rail infrastructure in the Netherlands. Nearly 4000 employees working in different departments construct, maintain and improve the tracks, organize the train schedules, manage the traffic and respond to incidents.

The construction and maintenance of the rail infrastructure are not performed by ProRail itself, but by different rail contractors depending on the region. Railway operators for passengers and goods pay a fee to ProRail for the use of railways. To facilitate the growth in travellers and goods, ProRail is constantly improving the tracks and railway processes to offer a sustainable mode of transport (ProRail, 2019a).

1.2 Problem introduction

This project focuses on the duration of railway incidents. An *incident* is defined as a negative, unexpected and unforeseen event that can be troublesome (van Dale, 2019). Railways incidents can have a big impact on train operations. Trains might have to be rerouted or cancelled, which results in hindrance to travellers and goods. In 2019, 209 high impact incidents occurred with more than 10 hours of accumulated delay each (ProRail, 2019b).

When an incident occurs, ProRail determines an *initial prognosis* of the incident end time. This initial prognosis and the information from the incident are used by the train traffic controller (TRDL, Treindienstleider) to inform the trains about the location of the incident. The railway operators use this information to replan their rolling stock and crew. The initial prognosis is also used to inform travellers and change train schedules.

Incident end time prognoses are made at different moments. The initial prognosis is used as an indication and it is given during the intake when the incident is reported, based on the information that is available at that moment. During the incident, new prognoses called *updated prognoses* can be provided by the general leader (AL, Algemeen Leider), who is the incident coordinator from ProRail. Updated prognoses keep the involved parties informed. Near the end time of the incident, a highly certain *final prognosis* is communicated by the AL, to make it possible to start replanning the train schedule.

To inform the TRDL, railway operators and travellers correctly, the prognoses of the incident end time must be reliable. Each under- or overprediction results in waiting time before the trains can start. Therefore, reliable prognoses result in less waiting time and better information for the travellers.

The initial prognosis is currently determined with a decision tree per incident type. The parameters for this decision tree have been determined based on previous research at ProRail. The updated and final prognoses are not determined with a decision tree, but given manually and are therefore largely based on the expertise of the AL. This results in differences at the moment in which the prognoses are given and differences in the reliability of these prognoses. ProRail believes that more reliable prognoses of the incident end time can reduce waiting time before trains can start and can improve the quality of the information to the travellers.

1.3 Definition of problem and goal of this project

Problems with incident end time prognoses and their relations are bundled in a problem cluster in Figure 1. In this section, the problems in the problem cluster are described and the problem statement and goal of this project are defined.

Many different types of incidents occur, each requiring a special process for recovery. A split between the incidents can be made between technical and non-technical. In the *technical* incidents, a contractor is required to perform a repair to the railway infrastructure. In a *non-technical* incident, other actions must be performed for incident recovery (e.g., when a train is malfunctioning, the train has to be pulled away).

During an incident, the AL supervises the incident recovery process, communicates with the different parties involved and makes decisions at the location of the incident. During an incident, the AL also gives new prognoses for the end time of the incident. At ProRail a final prognosis is considered *reliable* if it is given *in-time* and if it is *precise*.

An *in-time* final prognosis makes it possible to create an overlap between the last work activities that must be performed to finish the incident, and the replanning of the crew and rolling stock to restart the train traffic. This overlap will reduce the waiting time before the start of the first trains after the incident. Currently, 35 minutes is set as the time needed for the replanning. Therefore, a final prognosis is *in-time* if it is given at least 35 minutes before the actual end time of the incident.

A final prognosis is *precise* if the last work activities of the incident are finished before the time of the final prognosis. When the activities are finished after the final prognosis, the plan made for the restart of the trains must be changed. This results in delay and unclear communication to passengers. The activities that are finished before the final prognosis are less problematic, unless they are finished far before the final prediction, because this results in additional waiting time.

To give a final prognosis that is both *in time* and *precise* is complex because multiple features influence the incident duration. A *feature* is an attribute of the incident, such as the location or severity of the incident. The problem cluster in Figure 1 shows that the influence that these features have on the reliability of the prognosis is currently not known.

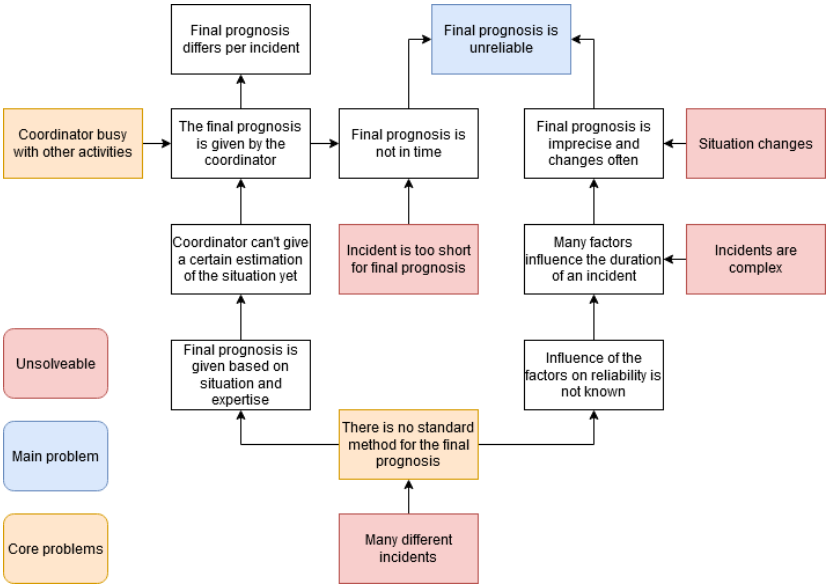


Figure 1. Problem cluster

In this project, *prognosis* is used for the prognosis given by the CQM decision tree and the AL. *Prediction* is used for the prediction generated by the data-based model developed in this project.

A data-based model for prediction that incorporates the influence of features on the incident end time, can provide more reliable predictions. A reliable prediction can be used to determine a prognosis which meets the desired level of overprediction. This would make the prognoses less depended on the expertise of the individual AL.

The problem statement for this project is defined as:

ProRail does not use a data-based method to support in time and precise prognoses for the incident end time.

Therefore, the goal of this project is:

Create a data-based model, based on literature and previous research at ProRail that provides predictions of the incident end time to support in time and precise prognoses.

To develop a data-based model, the features that influence the reliability of the predictions will be identified by data analysis of the current situation. The model will be tested for one incident type to contribute to the question of whether a more reliable prediction of the incident end time can be determined using a data-based prediction model.

1.4 Methodology

The Managerial Problem-Solving Method (MPSM) has been shown to be a useful method for solving business problems systematically while being able to include creative journeys to identify new and valuable alternatives (Heerkens & Winden, 2017). To solve the problem systematically, the MPSM consists of the following 7 steps (Heerkens & Winden, 2017):

1. Defining the problem
2. Formulating the problem approach
3. Analysing the problem
4. Formulating (alternative) solutions
5. Choosing a solution
6. Implementing the solution
7. Evaluating the solution

These steps serve as a guideline. It is also possible to make a loop and return to previous steps if a review is required. The structure of this report, outlined in Figure 2, follows the steps of the MPSM.

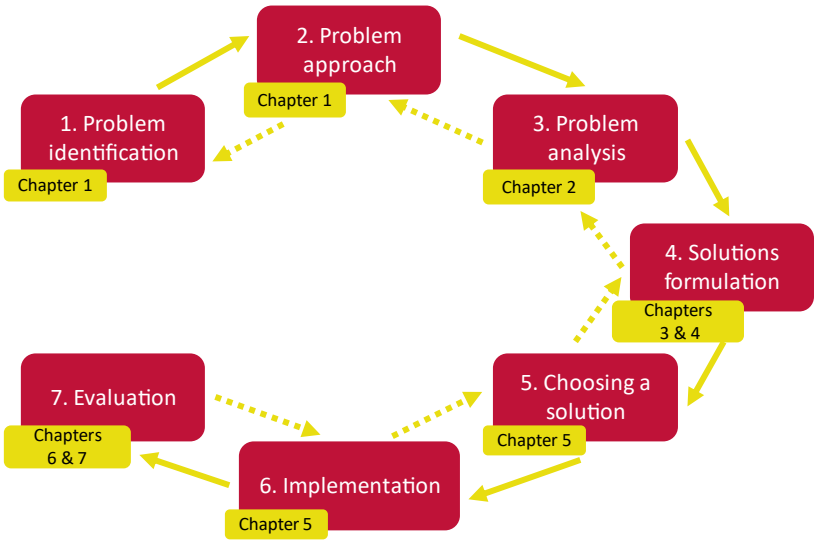


Figure 2. Report Structure

1.5 Research Framework

To solve the main problem, research questions are defined. These research questions structure the research process and are the questions that this project aims to answer.

1.5.1 The current process for incident recovery

Before focussing on incident end times, understanding the process of an incident, which parties are involved, and which steps are taken is important. Afterwards, it is also necessary to understand the process of prognoses and to define what makes a prognosis reliable. An analysis of the current reliability identifies the status and where an improvement is possible.

Based on the reliability and the impact on train operations, one incident type will be selected to focus on in this project. This results in the following five research questions:

- How is the process for incident recovery organized?
- How are incident end time prognoses currently determined?
- How reliable is the final prognosis currently?
- For which incident type does an improvement in the reliability have the biggest impact on train operations?

1.5.2 Previous research on incident duration prediction

To develop a data-based model for incident end time prediction, knowledge from previous research that has been performed at ProRail and internationally studies on incident duration prediction has to be considered. This leads to the following research questions:

- What previous research on incident duration prediction has been performed at ProRail?
- What research about incident duration prediction has been performed internationally?

1.5.3 Modelling methods

The aim of this master thesis is to build a model that can be used to provide a more reliable prognosis of the incident end time. From the prediction methods described in literature, the best method will be selected for incident end time prediction. To select the best method, performance measurement metrics and validation techniques to evaluate the performance are necessary. This results in the following research questions:

- What methods are proposed in the literature to develop a model for incident end time prediction?
- What metrics can be used to evaluate the performance of the identified methods?
- How can the performance of the developed model be validated and measured?
- Which method has the highest performance for the selected incident type?

1.5.4 Model development and performance

The best performing method for prediction will be used to develop a model for incident end time prediction. This model is applied during an incident to determine the prediction performance and which features influence the prediction. This leads to the following research question:

- What is the prediction performance of the developed model during incidents?
- Which features are important for prediction during incidents?
- Does the model support a more *in time* and *precise* prognosis?

1.6 Research scope

The scope of this project is the prediction of the end time of an incident during the incident. The initial prognosis will not be researched, because previous studies at ProRail (see Section 3.1) showed that minimal improvement is possible in the reliability of the initial prognosis. Interviews with an AL and other employees at ProRail working on prognoses suggested that the reliability of the final prognosis can still be improved.

Incidents that have an initial prognosis less than 60 minutes are not considered because a final prognosis cannot be given *in time*. This is because after the incident has been reported time is needed to assign an AL to the incident and for the AL to communicate with the TRDL to receive more information to base a new prognosis on. Therefore, if the prognosis has to be given 35 minutes before the end of the incident, the incident should be at least 60 minutes.

This project will focus on data-based decision support. The solution should serve as a support to the AL to make more substantiated decisions. The working routine of the AL will be used as a fixed process and will not be in the scope of this project.

2 Incident recovery process

In this chapter, the ideal incident recovery process from the start of an incident to the restart of trains is explained in Section 2.1. Then, the impact of unreliable prognoses is described in Section 2.2. In Section 2.3, remarks on the current incident recovery process are outlined. An analysis of the current reliability of the prognoses is performed in Section 2.4.

2.1 The ideal incident recovery process

The bathtub model can be used to represent the train traffic level during an incident (Ghaemi et al., 2017). This model consists of three phases for train traffic (see Figure 3). The first phase starts with the intake. The *intake* is when an incident is reported. Based on the information that is available from the intake, trains to the incident location get cancelled or are instructed to change tracks. The second phase starts after the schedules of trains to the incident location are adapted and ends when the plan to restart the train schedule is ready. The third phase starts with executing the restart plan and ends when the trains at the incident location are driving according to the original schedule again.

Throughout the incident recovery process, prognoses about the incident end time are made. There are three types of prognoses: initial, updated and final. The time a prognosis is determined will be referred to as the time a prognosis is *given*. In this section, the prognoses are expected to be perfect. In practice, this is not always the case. These situations and their effects on the incident recovery process will be described in Section 2.2.

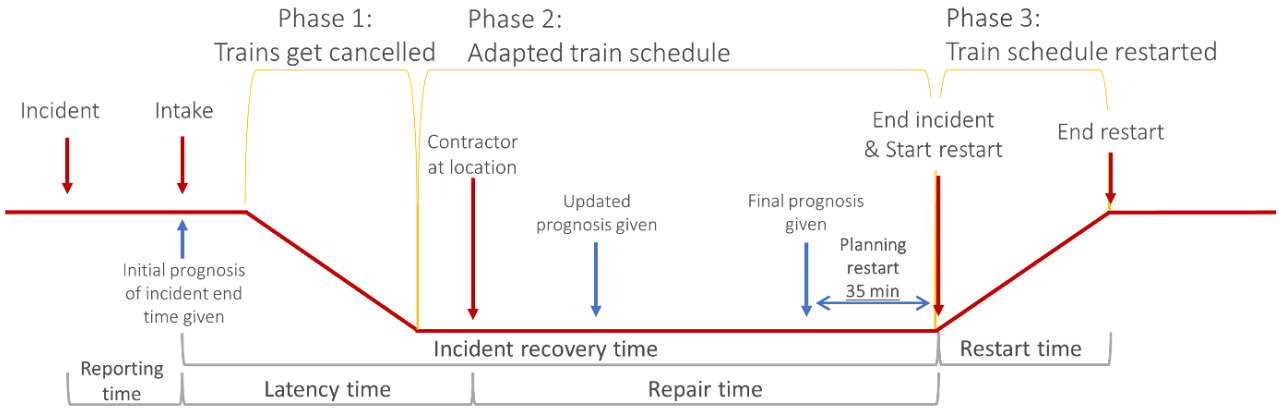


Figure 3. Incident recovery process bathtub model

2.1.1 Intake

The incident recovery process starts when an incident is reported to the Railway Alarm Room (MKS, Meldkamer Spoor). This is the central point at which all information about the incident is collected. With the available information that the reporter has of the incident, an intake form is filled. Based on this information, the MKS has to decide about the urgency, schedule a time for repair, alert government emergency services and determine the initial prognosis. The initial prognosis serves as a first indication to the railway operators and the travellers to act upon.

In the first phase of the bathtub model, the trains on or next to the track where the incident occurred have to be cancelled, redirected or instructed to drive slowly. This is the responsibility of the TRDL. The initial prognosis is used by railway operators to communicate the travel information with their travellers and change the planning for rolling stock and crew. Currently, railway operators decide to cancel trains until the time of the initial prognosis.

The initial prognosis is given in SpoorWeb, the information system of ProRail for handling incidents, by a decision tree made by the consultancy company Consultants in Quantitative Methods (CQM). For every incident type, both technical and non-technical, a decision tree is constructed (see example in Figure 4). Based on the incident type, features of the incident are selected to create a split in the data. For a split, the feature is selected for which the 65th-percentile of the distributions of the options of the feature are not within the 95% confidence interval range of the other options. For example, for the feature rain the 95% confidence interval of "no" is [45,49] and for "yes" is [39,53]. The 65th percentiles are 47 and 45 respectively. Because the 65th percentile of "no" (47) is within the confidence interval of "yes" [39,53], this feature is not used for a split. If multiple features can be used for a split, the feature is selected with the highest difference in duration for the leaves of the tree. The tree stops at the leaf in which no features satisfies this criterium (CQM, 2019). The decision tree is implemented in SpoorWeb and automatically gives the MKS an initial prognosis during the intake based on the features of the incident that are entered in the intake form.

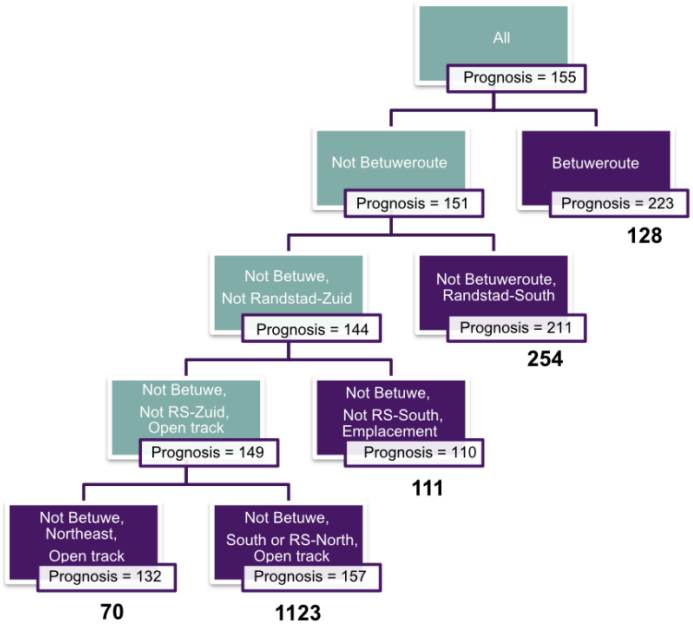


Figure 4. Decision tree section malfunction (CQM, 2019)

The initial prognosis is the 65th percentile of the distribution at the leaves of the decision tree. The 65th percentile is used for the initial prognosis because a pessimistic expectation is better than an optimistic expectation. An optimistic expectation can result in an underprediction of the incident duration. This leads to prognoses that often have to be extended, which makes them unreliable. The 65th percentile means that 65% of the incidents with the same features have been finished before the time of the initial prognosis. For the construction of the decision trees, multiple data sources from 2014 to 2018 were used.

2.1.2 Incident recovery

After the intake, an AL is assigned to the incident. The AL can update the expected incident end time with an updated prognosis, e.g., when reading the intake form or when receiving new information from the assigned contractor or other parties. When the incident is considered to have a high impact, the MKS requests the AL to go to the location of the incident and the TRDL starts instructing or changing the trains that are affected by the incident. When all the trains on the track where the incident occurred or on the tracks next to the incident track have received driving instructions or are changed, phase 1 of the bathtub model is ended. In phase 2 the affected trains drive according to an adapted schedule to ensure the safety of the people at the location of the incident.

The parties involved with the incident differ per incident type. For technical incidents, a contractor has to come and perform repairs. For other incidents, the police or fire department might be involved. With incidents that involve a stranded train, the incident response team of ProRail also goes to the location and takes care of the passengers and the train.

From the moment the AL is assigned to the incident, the AL communicates with the parties involved in the incident recovery process, determines the actions that have to be performed and records the duration of these actions. An AL can decide to update the expected incident end time prognosis with information gained during the incident with his expertise and experience.

With the current model of CQM, it is not possible to use new information to determine updated prognoses. This is because the decision tree of CQM is static. This means that the decision tree is the same for every incident of that type and based only on information at the intake. A dynamic tree could consider information after the intake. As new information becomes available, a new prognosis could be made with a dynamic tree that is specific to the incident. In this project, the decision tree of CQM will only be used to identify important features of incident types.

2.1.3 End of the incident and after the incident

At the latest 35 minutes before the expected incident end time, a new prognosis for the expected incident end time should be given by the AL. This can be an updated prognosis if the duration of the incident is uncertain, or a final prognosis when the AL is highly certain that the incident will be finished by that time. The moment in which a final prognosis is given should be at least 35 minutes before the end of the incident for a prognosis to be *in time*, because this time is needed for the replanning of the rolling stock and crew by the railway operators and traffic control (VL, verkeersleiding). The effects of a prognosis that is not given *in time*, or the effects when the prognosed incident end time is not correct, will be explained in Section 2.2.

The moment in which an incident is finished, the AL marks the end of the incident recovery activities. This is called *End ICB (Einde Incidenten Bestrijding)*. At this moment, phase 2 has ended and this is the signal for the TRDL to allow trains to start driving according to the restart schedule at the location of the incident, phase 3. When all the trains are back to the original schedule, the restart is completed and phase 3 is ended. An overview of the actions by the different parties is displayed in a swimlane diagram in Figure 5. All parties involved in the process and their roles are summarized in Table 1.

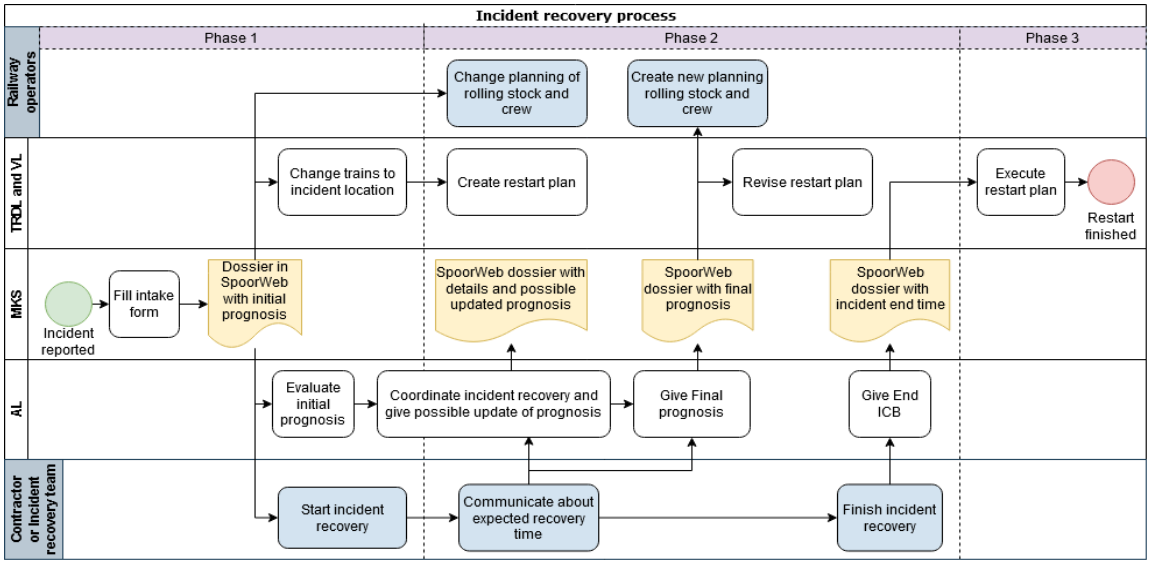


Figure 5. Swimlane diagram of the incident recovery process

Table 1 Tasks per role in the incident recovery process

Role	Tasks
MKS	Central point of contact, fill intake form, generate initial prognosis, inform involved parties.
TRDL and VL	Cancel, redirect, or instruct trains when an incident occurs and restart the train traffic after the incident.
Railway operators	Communicate travel information, change planning rolling stock and crew.
AL	Coordinate incident recovery process, update prognosis, and give a final prognosis.
Incident response team ProRail	Take care of passengers and stranded trains.

2.2 Unreliable prognoses

A final prognosis that is not given *in time* can result in a delay before trains can be restarted because the restart plan is not finished when the incident is finished. Figure 6 shows the delay between the actual end of the incident and the start of the restart.

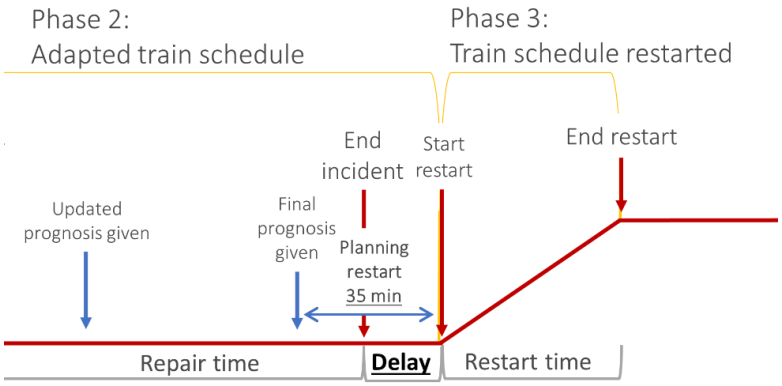


Figure 6. Final prognosis not *in time*

If the prognosed incident end time is before the actual incident end time, the prognosis is called as *precise*. When the actual end time of the incident is later than the prognosed incident end time, the restart plan has to be adapted because the incident is not resolved, and trains cannot start driving at the expected time. Then, the planners will wait till the end of the incident before creating a new restart plan. Thus, a prognosis that is not *precise* will result in a delay after the incident is resolved (Figure 7).

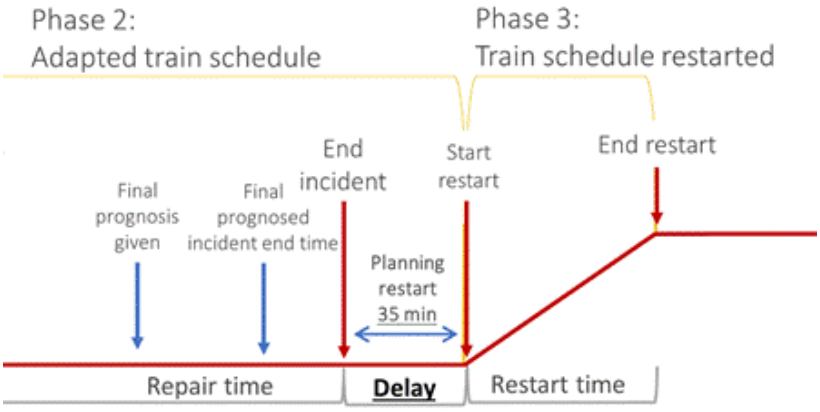


Figure 7. Prognosed end time before the actual end time

If the actual incident end time is before the prognosed incident end time, the restart plan does not have to be changed. However, the time between the actual incident end time and the prognosed incident end time is additional waiting time until the restart begins (Figure 8). Therefore, a prognosis that overestimates the incident end time, a pessimistic prognosis, can lead to additional waiting time.

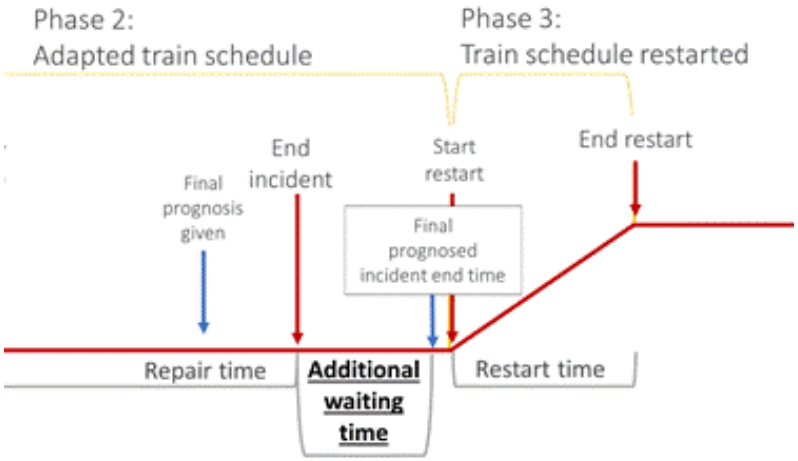


Figure 8. Actual incident end time before prognosed end time

The goal is to have a final prognosis that is given *in time*, (i.e. 35 minutes before the actual end of the incident), and *precise* (i.e. the prognosed end of an incident is not before the actual end of the incident). In practice, providing a reliable final prognosis is difficult. In Section 2.4, a preliminary analysis is performed to identify how *in time* and *precise* the prognoses currently are.

2.3 Remarks about the incident process

The initial prognosis is the time that follows from the leaves of the CQM decision tree. This expected time is a point and does not communicate the distribution of the end time of the incidents. For the parties involved, the initial prognosis serves as an indication of the time to aim for. Because the initial prognosis is based on the 65th percentile instead of the median of the distribution of duration of incidents at the same leaves of the decision tree, the AL's are in 65% of the cases aiming for an end incident time that is too long.

Data stored from the actions that are performed at the incident location for technical incidents is limited. This is due to the employment of contractors. Contractors store the causes of the incident in their own databases, which are not shared with ProRail. Another reason for minimal data is that logging performed actions takes extra time and could cause more delay.

Prognoses currently change often when new information becomes available, (e.g., a repair that takes longer than expected, or when the identified cause doesn't solve the problem). When an incident is resolved shortly after the time of the current prognosis, the planners have to change the planning for the restart again. In multiple interviews, it was stated that, currently, because of this planners have to wait until the incident is solved completely. This means that new changes cannot occur before they start planning the restart. This results in a delay with the duration of the restart planning process.

To avoid frequent changes of the restart plans, the railway operators currently cancel trains during the incident until the longest known prognosis: initial, updated or final prognosis. Since the initial prognosis is pessimistic, it should only be used as an indication and not as a fixed time. Reliable final prognoses would make it possible for the railway operators to focus more on the final prognosis instead of longest prognosis. This would change the role of updated and final prognoses in the recovery process and could lead to less additional waiting time before trains can start driving after an incident.

2.4 Current reliability of final prognoses

The incident type of an incident is determined during the intake. In this section, an analysis of the current reliability of the final prognosis is performed. This analysis will compare the differences in reliability per incident type.

2.4.1 Data analysis for problem identification

Data from 1st of January of 2020 to the 30th of September 2020 is used for the problem identification analysis. The data includes (1) the times the prognoses are given, (2) the prognoses times themselves and (3) the time in which the incident is resolved.

Incidents are first filtered on an initial prognosis of ≥ 60 minutes because incidents that have an initial prognosis of < 60 minutes are out of the scope of this project.

When no final prognosis is given manually, the system will automatically give a final prognosis at the moment the incident is marked as resolved in SpoorWeb with that time. To accurately analyse how much of the prognoses are *in time* and *precise*, the incidents with the automatic final prognosis are filtered out.

2.4.2 Overview analysis

This analysis gives an overview of how *in time* and *precise* the prognoses of the 9 most occurring incidents are. When a prognosis is changed (updated or final), the corresponding value in the database is overwritten. Therefore, the following timelines of the prognoses of an incident are based on the last stored prognoses.

In time

A final prognosis is *in time* if the time the final prognosis is given is 35 minutes or more before the end time of an incident. Thus, a prognosis is *in time* if [End incident] - [Final Prognosis Given] ≥ 35 minutes. An overview of how *in time* the prognoses are for the 9 most occurring incident types is shown in Figure 9.

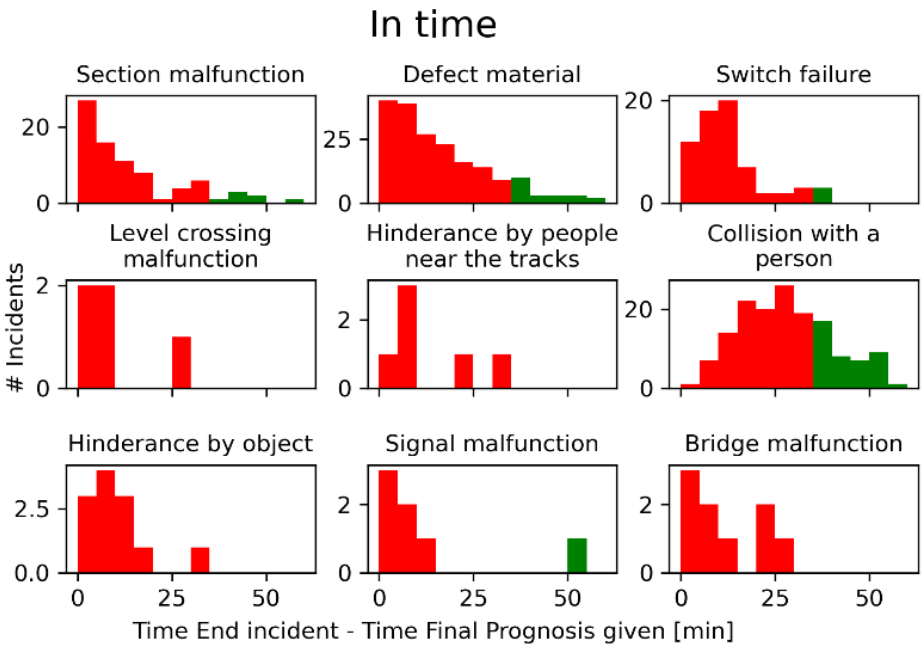


Figure 9. Overview incident types *in time* prognoses

The *in time* overview shows that for most incident types, the final prognoses are not *in time*. Some final prognoses are given more than 50 minutes before the end of the incident, but most are given less than 35 minutes before.

Precise

A final prognosis is *precise* if the predicted final prognosis time is equal to or greater than the incident end time. So, a prognosis is *precise* if [Final prognosis] - [End incident] ≥ 0 minutes. An overview of how *precise* the prognoses for the 9 most occurring incident types are shown in Figure 10.

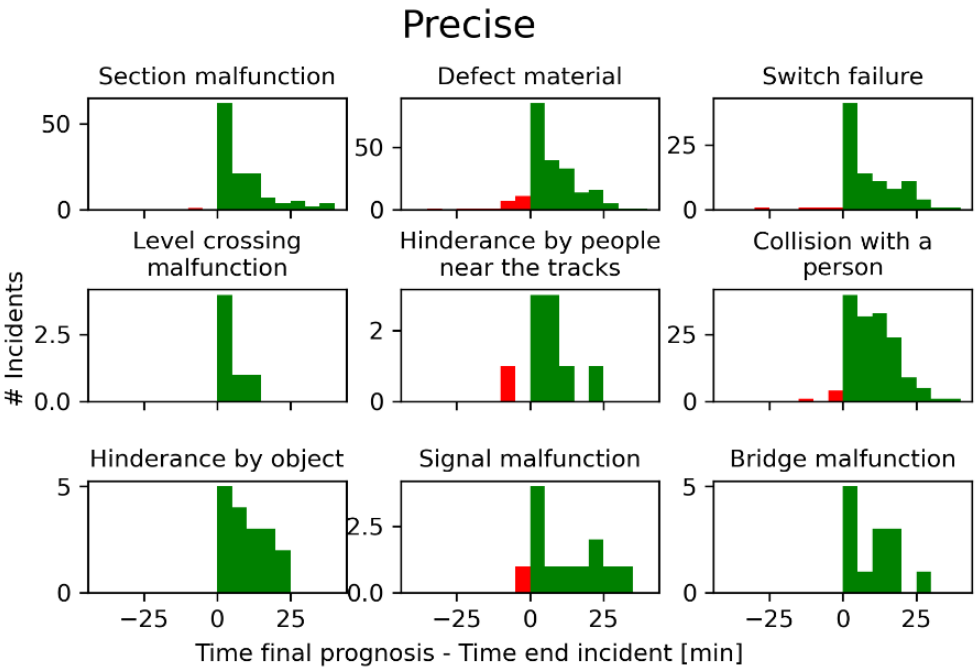


Figure 10. Overview incident types *precise* prognoses

The *precise* overview shows that almost all of the prognosed incident times are after the actual incident end time. The green area shows the additional waiting time, which is between 0 and 30 minutes for all incident types. This means that the final prognoses give an overprediction of the actual time that is needed till the end of the incident. According to the definition at the beginning of this paragraph, a final prognosis that is past the actual end of the incident is called *precise*.

2.4.3 Individual incident types

The incident types defect material and the collision with a person will be compared here. These incident types were marked by ProRail as the most interesting incident because they occur often and do not involve a third-party contractor. The incident type for which a model will be constructed will be selected in Chapter 4.

Defect material

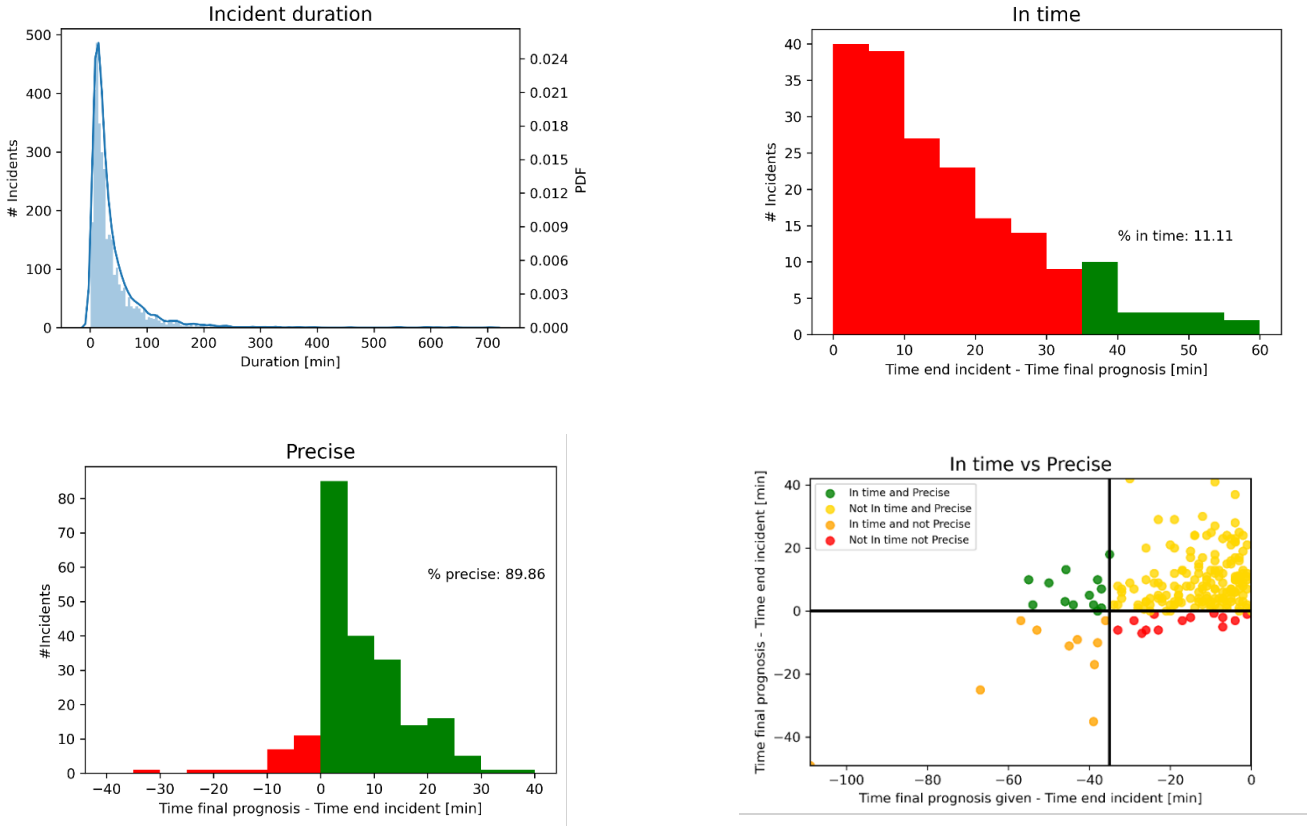


Figure 11. Defect material plots.
Top left: Distribution duration. Top right: *In time*. Bottom left: *Precise*. Bottom right: *In time* vs *Precise*

From the duration plot of defect material in Figure 11, it can be seen that most incidents are shorter than 100 minutes. The *in time* plot shows that only 11% of the prognoses of the incidents longer than 60 minutes is given *in time* and most final prognoses are given just before the end of an incident. The *precise* plot shows a 90% overestimation and a peak between 0 and 5 minutes. This means that the final prognosis was very *precise*. From the *in time* vs *precise* plot, it can be seen that most prognoses that are *precise* are not given *in time*. From discussing these plots with people from ProRail and reading logging information about defect material incidents, it becomes clear that only shortly before the end of an incident enough information is available to predict the incident end time.

Collision with a person

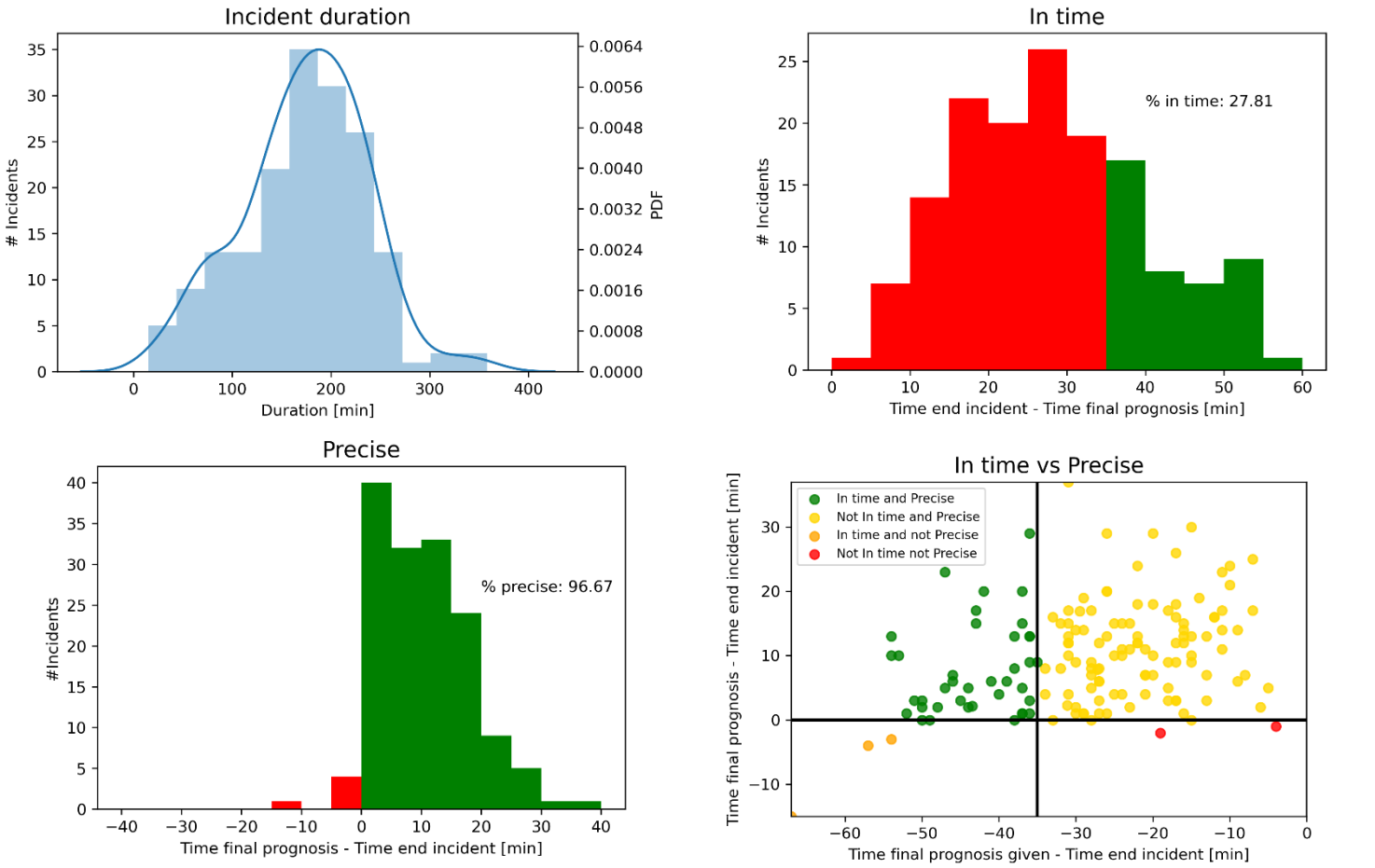


Figure 12. Collision with a person plots
Top left: Distribution duration. Top right: *In time*. Bottom left: *Precise*. Bottom right: *In time* vs *Precise*

The duration plot in Figure 12 shows that the duration of a collision with a person incident is more distributed compared to defect material. Most collision with a person incidents last between 2.5 and 4 hours. From the *in time* plot, it can be seen that only 26.5% of the prognoses are given more than 35 minutes before the incident end. The final prognoses are *precise*, since almost all incidents have an overestimation of 10 to 20 minutes, as indicated by the green area in the precise plot. The *in time* vs *precise* plot shows once again that most of the final prognoses are *precise* but are not *in time*.

2.5 Summary

In this chapter, the incident recovery process is explained with the parties involved. In the process, a good estimation of the end time is crucial to minimize the waiting time before trains can start running again. Currently, only the initial prognosis is given based on a unique decision tree per incident type. Based on the features of an incident, the 65th percentile of the distribution at the leaf of the tree is taken for the initial incident duration prediction. During the incident recovery process, an updated prognosis can be given by the AL. When the incident end time is highly certain, a final prognosis is given. The reliability of this final prognosis depends on the factors *in time* and *precise*. An analysis of the factor *in time* for the most occurring incident types showed that most prognoses are given less than 35 minutes before the end of the incident. So, they are not given *in time*. Most of the predicted incident end times were after the actual end times. The analysis shows relatively high overestimation, which results in extra waiting time. The analysis of incidents concerning defect material and collision with a person shows that for defect material incidents the final prognoses are given at the very last moment and, therefore, these are also *precise*. For collision with a person, most prognoses are given more than 10 minutes before the end of the incident. The analysis showed an overestimation of 10 to 20 minutes, which results in additional waiting time. This shows that the reliability of the prognoses can still be improved. In chapter 3, a literature review will be performed to identify models to make a reliable prediction of the end time of an incident.

3 Literature review

In this chapter, a literature review is performed. First, previous research at ProRail about prognoses is described in Section 3.1. In Section 3.2, methods for selecting features, to be included in a prediction model, are described. Methods proposed in the literature to develop a model for incident time prediction are discussed in Section 3.3. This chapter ends with model validation in Section 3.4 and a summary in Section 3.5.

3.1 Previous prognosis research at ProRail

Four research projects about prognosis have been performed at ProRail from 2015 to 2019. In this section, the focus, methods and outcomes of these projects are summarized.

De Wit (2016) focused on the initial prognosis. Four methods for initial prognosis were proposed: confidence intervals of probability distributions, regression analysis, nearest neighbour and prediction by an expert. For these methods, features were determined that result in a more accurate prediction. To communicate the reliability of a prediction, confidence intervals were proposed. Probability distributions and regression analysis showed the best prediction performance. The average success percentage for 25-minute intervals was only 5 percent off from the actual time.

The projects of Zilko (2017), DataLab ProRail (2019) and Wemelsfelder (2019) at ProRail focused on both the initial and updated prognosis. After an initial prognosis was determined, an improved updated prognosis was supported when new data became available.

3.1.1 Technical incidents

Zilko et al. (2016) proposed a Bayesian Network model to predict the length of an incident based on the statistical dependencies of variables. This model can give a prediction for the incident duration when information is still missing. When new information becomes available, the distributions are updated and this results in a new prediction.

The length of an incident was split into the latency time and the repair time. Features that influenced the latency time were time, location and weather. Features that influenced the repair time were contract type and the cause. An example model was created and resulted in a better prediction compared to the initial prognosis. The model represented the data well, however, the R^2 was low. They concluded that the data used was of poor quality and that expanding the model with more influential features could have a potential benefit (Zilko, 2017).

In 2019 the DataLab of ProRail focused on the initial prognosis of section malfunctions (DataLab ProRail, 2019). The project attempted to determine the cause of the incident with text mining. As also identified by Zilko (2017), the cause influences the incident duration. With the results from text mining, new decision trees were constructed with the features: time, location, contract type, Train Incident Scenario (TIS, Trein Incident Scenario), equipment type and the cause.

These decision trees showed that the distribution of incident end time changes per cause, but still had large deviations. To inform about the uncertainty of a prognosis, the project proposed to communicate the point estimate at the 65th percentile and also the 35th and 85th percentile of the prediction distribution.

The impact of new information during the incident on the width of the prediction distributions intervals was also investigated. This showed that the later a prediction is given, the higher the certainty. The recommendation from this project was to identify moments when new information becomes available to give a new prognosis. However, the project showed that the incident recovery process is difficult to predict. New prognoses still have uncertainty, for which intervals can be a good method of communication.

3.1.2 Non-technical incidents

The research of Zilko (2017) and DataLab ProRail (2019) was limited to technical incidents. Therefore, Wemelsfelder (2019) researched a dynamic model for prognosis that can determine an updated prognosis when new data becomes available, even for non-technical incidents. The methods used were Bayesian Networks (BN) and k-Nearest Neighbour (kNN). A decision tree was also identified as a suitable method but was excluded because the actual CQM decision tree (see Section 2.1.1) was already a decision tree. The feature selection for the model of Wemelsfelder combined features from Zilko, De Wit and CQM. As an example, features selected for three incident types are displayed in Table 2.

Table 2 Features per incident type (Wemelsfelder, 2019)

Rolling stock	**Section TOBS**	**Collision/Hindrance**
HSL/Betuwe	Day/Night	Randstad
Driving characteristics	Working hours	Working hours
Rolling stock type	Randstad	Day/Night
Freight train	Contract type	Thing train collided with
Day/Night	Temperature	Nature of incident
TIS	Overlapping incidents	Location of base
Train table adjusted	Rush hour	Train table affected
Train company	Year of replacement	
Shunting point	Location of base	
Activity	Contractor	
	Tao indicator	
	Wind direction	
	Cause	

The performance of the kNN and BN model was measured with RMSE and MAE (see Section 3.7.1) and showed similar performance to CQM for the initial prognosis. For the updated prognosis only the performance of the collision/hindrance incidents improved. This evaluation has however been performed on a small number of data points, 20 and 10 respectively. An analysis where extra time was added to the prediction, showed that the impact of overprediction on the prediction errors was minimal. Therefore, adding time to the predicted time from the model would decrease the probability of underpredicting and results in a minimal increase in prediction errors.

3.2 Machine learning

Studies in China (Huang et al., 2020), Sweden (Corman & Kecman, 2018; Nilsson & Henning, 2018), Denmark (Grandhi, 2019) and the Netherlands (Wemelsfelder, 2019; Zilko, 2017) showed the use of Machine Learning (ML) models for prediction in railway. ML can help solve problems that are complex and contain large amounts of data (Mehryar et al., 2019).

The goal of ML is to find a balance between bias and variance. Bias is when the model cannot capture the complexity of the real-life situation. Variance is the amount that the prediction changed when different historical data is used. Obtaining low bias and low variance is the goal. Before explaining the models used, the different types of ML models will be presented.

There are three types of ML: (i) supervised learning, where historical data is used to train the model to predict the output variable, (ii) unsupervised learning, where the output variable is unknown and the model has to find the structure on its own (Hastie et al., 2008), (iii) and reinforcement learning, where the actions to maximize a reward have to be found. The model provides no answer but has to decide the actions to perform itself (Abu-Mostafa et al., 2012).

Because of the previous use of machine learning methods for prediction, only machine learning methods will be researched. Other, more statistical, methods can also be used but will not be researched in this project.

3.2.1 Machine learning process

For this project, previous incidents will be used for prediction. Therefore, this project focusses on supervised learning methods. The process of developing a supervised machine learning model consists of multiple steps (Akinsola, 2017). An overview of these steps is given in Figure 13.

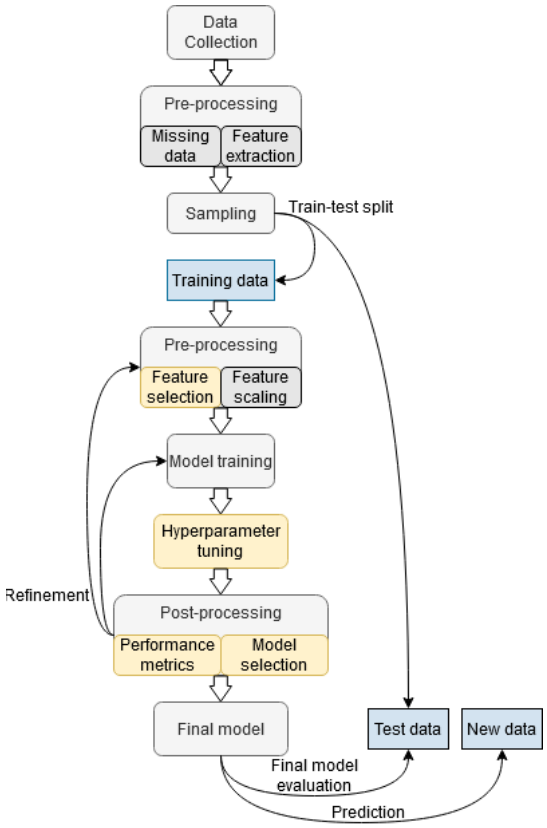


Figure 13. Supervised machine learning process

Methods for feature selection will be researched in Section 3.3. Different models for prediction in railway and hyperparameter tuning will be researched in Section 3.4. Performance metrics to compare models for selection will be explained in Section 3.5. The other steps of the supervised machine learning process will be discussed and applied to the incident data in Chapter 4.

3.3 Data pre-processing

Data pre-processing is one of the very important steps because data is prone to a lot of anomalies, missing information and inconsistencies. Data pre-processing aims at improving the quality of raw data and, consequently, the quality of mining results. It also prepares the data to enable further analysis (Jambhorkar & Jondhale, 2015).

After the data collection, first, the data has to be analysed for anomalies and inconsistencies. When the data is cleaned, missing data can be dealt with.

3.3.1 Missing data

Some machine learning methods can deal with missing data. When a machine learning model is not able to deal with missing data, the data has to be deleted or imputed. There are three common reasons why data is missing in a data set:

1. **Missing Completely at Random**
   The data that is missing has no relation with the other data in the data set.
2. **Missing at Random**
   The missing data can be explained by data from other features.
3. **Missing not at Random**
   The reason why the data is missing is related the data itself.

When the data is missing completely at random, or at random, it can be deleted without impacting the bias of the model. For data that is not missing at random, deleting data would increase the bias of the model. In this case, imputation can be an option (Allison, 2001).

The imputation method depends on the type of data. If the data is categorical, some imputation methods are adding missing as a category, select the most frequent category, or use prediction models to predict the missing values. For numerical data, mean, mode, median or regression can be used to impute the missing data. k-Nearest Neighbour can also be used for both categorical and numerical data to impute the data with the average of the neighbours.

Imputation can result in higher bias because imputed data may be too similar to the other data. Also, imputation does not have to lead to better results than deletion, but when data is sparse and the model cannot deal with missing data, it can be useful.

3.4 Feature selection

Selecting which features to include in the model is an essential step in the model creation process. Feature selection can result in better model performance by reducing overfitting, decreasing the training time and providing an understanding of the data. There are three different methods for feature selection: filter methods, wrapper methods and embedded methods (Guyon & Elisseeff, 2003). These three methods can be used both separately or combined for a more robust model selection.

Filter methods are statistical techniques to evaluate the relationship between features and the output variable. Filter methods mostly compare one feature with the output variable. Because of this, the interaction between features is not evaluated. When the features are numerical, correlation techniques like Pearson's correlation can be used. When the features are categorical, ANOVA can be used (Kuhn & Johnson, 2013).

Wrapper methods use a different machine learning algorithm in the core of the method and its performance is used as an evaluation method to select features. Many models are created that add or remove features to find the combination of features with the best performance. Common wrapper methods are forward selection, backwards elimination and stepwise selection. *Forward selection* starts with an empty model and adds features that result in the highest increase in the performance measure. *Stepwise selection* makes forward selection less greedy by revaluating all features in the model for elimination after a feature is added. *Backward elimination* starts will all features and removes features that results in the smallest decrease of the performance measure.

Recursive Feature Elimination (RFE) performs a greedy search by iteratively removing features from the model and creating models on the remaining features. The feature that showed the lowest performance is removed. When all features are evaluated, feature ranking is giving by the order of elimination.

Embedded methods perform the feature selection during the training of a model. Common embedded methods are regularization methods that penalize additional features. During the optimization, constraints penalize extra features leading to a higher bias model with fewer features and variance.

There are two types of regularization, L1: *Lasso Regression* and L2: *Ridge Regression*. Lasso Regression penalizes the absolute value of the magnitude of the feature where Ridge Regression penalizes the square of the magnitude of the feature. Lasso Regression can shrink the coefficients of features to 0, where Ridge regression uses all features in the model. Therefore, Lasso regression excludes useless features where Ridge regression is better when most features are useful (Hastie et al., 2008).

Other embedded methods are tree-based methods (see Section 3.5.2) which calculate feature importance during the training of the model.

3.5 Modelling methods

In the next two paragraphs on modelling and evaluation methods for prediction, both projects on incident duration and train delay prediction are reviewed, because the prediction methods used in these project are very similar (Ghofrani et al., 2018).

Two types of variables can be predicted: discrete and continuous variables. The prediction of a discrete output variable is a classification. The prediction of a continuous output variable, such as the duration of an incident, is done with regression.

3.5.1 Linear Regression

The simplest type of regression model is a linear regression. In linear regression, the output variable is predicted by the linear combination of the input variables with weights. The weights are optimized to minimize the squared error between the prediction and the actual value of the output variable. Linear regression is built on the assumption that there is a linear relationship between the input variables and the output variable. When there is no linear relationship between these variables or unequal variance across the variable (heteroscedasticity), linear regression shows low performance (Hastie et al., 2008).

3.5.2 Tree-based models

Trees are used to partition the feature space in groups. For partitioning, the feature with the least reduction in accuracy is selected. To prevent a tree from splitting on too many features which results in overfitting, a minimal number of observations in a group can be set, or a maximum number of splits can be defined (Hastie et al., 2008). Advantages of decision trees are that they are easy to interpret by users and that nonlinear relationships of features do not influence the performance.

Ensemble learning is the combination of multiple individual models which combined give a more accurate model. For instance, *Random Forest* is an ensemble of decision trees. With Random Forest, many decision trees are constructed in parallel based on subsets of the dataset. The subsets are created randomly by selection with replacement, this is called *bootstrapping*. The available features differ per tree due to the randomness of the bootstrapping. The final prediction of the Random Forest is determined by averaging the predicted values of the trees. Since many random trees are built, a Random Forest is resistant to overfitting and the accuracy is higher compared to decision trees (Breiman, 2001).

Another method to build forests is *boosting*. In a boosting method, subsets are selected from data sequentially. A first subset is selected randomly and points that have low prediction performance are included in the next sample with new random points. This helps models to improve wrongly predicted points by focusing on them. However, this can increase overfitting and variance (Hastie et al., 2008).

Different boosting techniques exist. *AdaBoost* and *gradient boosting* are the most common. In AdaBoost weights of each tree can be different and trees are based on the error in previous trees. AdaBoost is used by Nilsson & Henning (2018) for train delay prediction and showed reasonable performance. *Gradient boosting* is a greedy method that sequentially selects trees at each step that minimize the loss function. *XGBoost* is an extension of gradient boosting and uses more regularized model formalization to control overfitting (Chen & Guestrin, 2016). Grandhi (2019) showed that XGBoost performed well for incident duration prediction on training data, but showed overprediction due to the data used, causing a lower test performance.

3.5.3 Bayesian Networks

A Bayesian Network (BN) is a directional acyclic graph. The nodes represent the random variables, and the edges correspond to the conditional probability of the nodes. A node first holds the probability distribution of the random variable independent of the other nodes. When the value of a random variable is known, the BN updates the probabilities of the connected nodes based on the conditional probabilities. Figure 14 shows the BN of Zilko (2017) where the contract type is known, but the cause of the incident is not yet known. Therefore, the independent probabilities or different causes are displayed and the distribution of repair time is determined based on the statistical dependency between these variables. When the cause is known, the distribution of the repair time changes and therefore the distribution of the disruption length as shown in Figure 15.

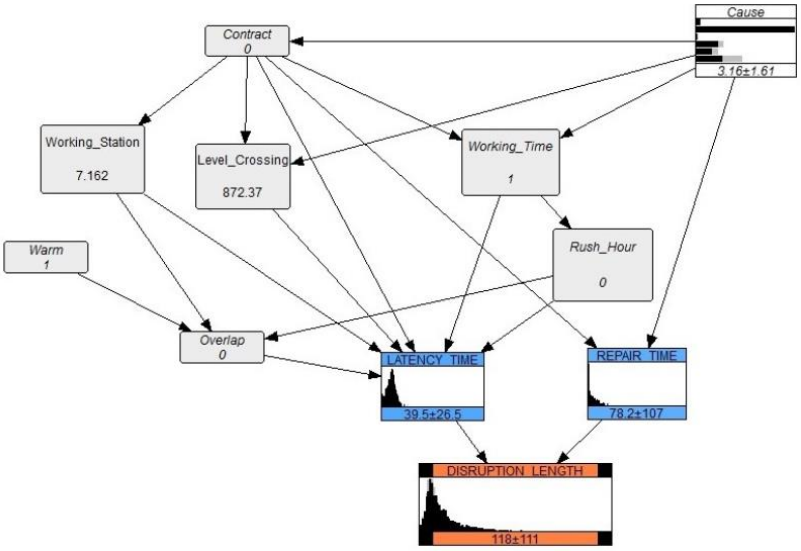


Figure 14. Bayesian Network with unknown cause (Zilko, 2017)

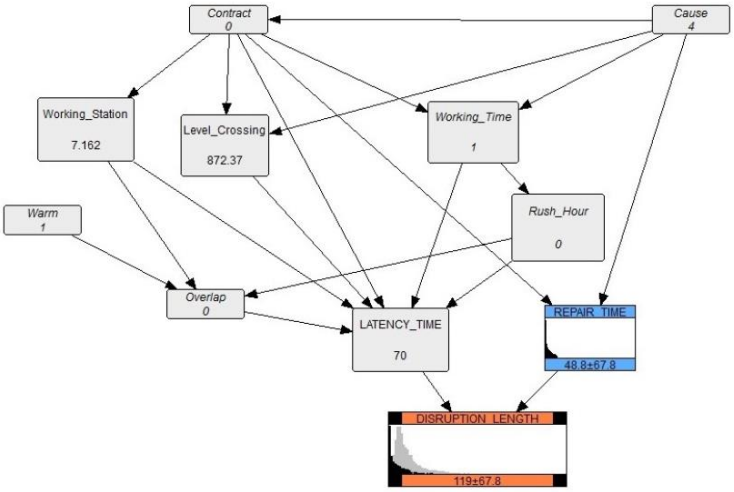


Figure 15. Bayesian Network with known cause (Zilko, 2017)

Corman & Kecman (2018) proved that a BN is an appropriated method to model the complex interdependencies with train delays. Lessan et al. (2019) showed that a BN, built with domain knowledge and experts' judgements, can achieve a prediction performance for train delays.

3.5.4 Artificial Neural Networks

Artificial Neural Networks (ANN) are inspired by the neural network of the brain. An ANN consists of layers and each layer consists of nodes. There is an input layer for the features, one or multiple hidden layers and an output layer for the prediction. The way nodes are connected between layers depends on the architecture and influences the ability of the nodes to retain information. The connections between the nodes have a weight and each node has a bias. The activation of a node is determined by an activation function. The activation function receives as input: the weighted sum of the connections and the values of nodes plus the bias. During model training, the weights are changed to minimize a loss function with gradient descent. This is called backpropagation (Hastie et al., 2008).

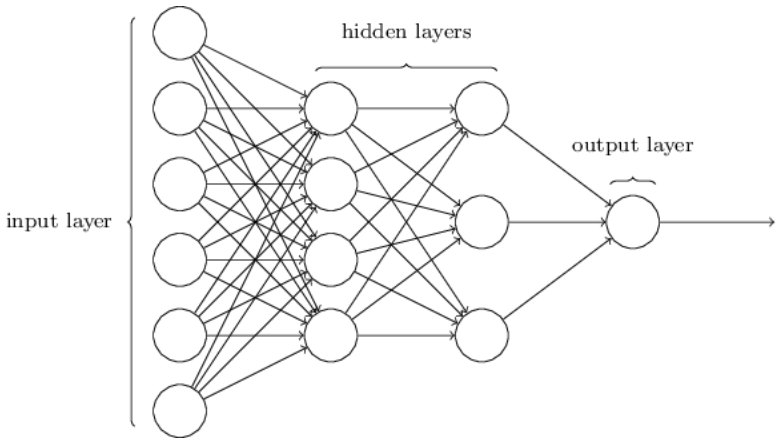


Figure 16. Example of an Artificial Neural Network (Nielsen, 2015)

An ANN can model non-linear relations and learn the relation between features in the data. An ANN can handle missing data by not activating a neuron. Because of the large amounts of weights that are changed, ANNs are not interpretable and are called block box. Before training the model on data, the number of layers, nodes, activation function, learning rate and stopping criterion has to be set and influence the training time of an ANN. There is no standardized method for these parameters (Nielsen, 2015).

An ANN with 3 hidden layers and 22 nodes per layer resulted in the lowest MAE for Nilsson & Henning (2018) for prediction of the duration of train delays in Stockholm. They argued that this network and a network with 2 hidden layers and 22 nodes per layer might be too complex and cause overfitting. Grandhi (2019) found that an ANN with 3 layers of 27, 5, and 5 nodes, showed consistent performance between the historical data and new data for duration prediction.

3.5.5 Support Vector Machine

Support Vector Machines (SVM) separate between classes of data points, by fitting a hyperplane in N dimensions. The objective of this hyperplane is to maximize the distance, also called *margin*, between the data points of the classes and the hyperplane. The support vectors are the points that are close to the hyperplane and determine the position and the orientation of the hyperplane. Support Vector Machines are mostly used for classification due to the nature of separating classes and use linear separation. Non-linear separation can be achieved with a kernel function which maps the non-linear observations to a (higher-dimensional) space, where there become linearly separable (Hastie et al., 2008).

The advantages of an SVM are that they are computationally efficient and can be used with high dimensionality datasets. Because the generalization is built into the model, they are more robust against overfitting. The disadvantage is that SVM models are difficult to interpret and the hyperparameters (See Section 3.6) are difficult to determine, which impacts the model performance.

SVM's are used by Valenti et al. (2010) and showed the best performance, compared to linear regression and ANNs, for predicting the length of medium to long incidents.

3.6 Hyperparameter tuning

Hyperparameters are model parameters that are set before training. By tuning the hyperparameters a higher model performance can be achieved. Both grid search and random search are methods for tuning hyperparameters. Grid search generates candidates using a predefined grid. The number of parameters to evaluate has a negative influence on the computation time. Random search selects parameters randomly. Random search is faster and adding parameters has less influence on the computation time compared to grid search. The performance of random search is equal to or better than grid search (Bergstra & Bengio, 2012). The difference between the two methods is displayed in Figure 17.

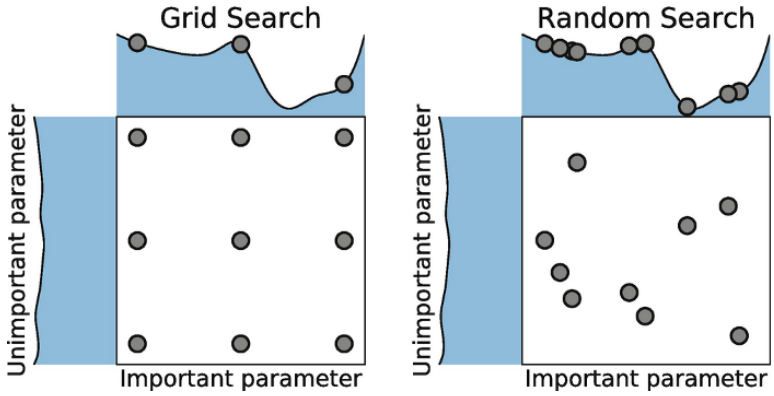


Figure 17. Grid search vs Random search for hyperparameter optimization

To ensure that the estimated parameters do not overfit the training data, the performance of the model has to be validated. This can be done with a validation dataset. To prevent the parameters being depended on the split in training and validation set, k-fold cross-validation can be used. With k-fold cross-validation, a random subset of the original training data is chosen to be the validation set and the other k-1 subsets are for training the model with the parameters. This process is repeated *k* times, every time using a different subset for validation. In the end, the results are averaged to give a good representation of the performance of the parameters (Hastie et al., 2008).

3.7 Model evaluation

In Section 3.5, tree-based models, bayesian networks, artificial neural networks and support vector machines are discussed and how hyperparameters for these models can be determined in Section 3.6. Common metrics to evaluate and choose regression models for prediction in railway, communicating prediction uncertainty and final model validation will be discussed in this section.

3.7.1 Evaluation Metrics

The Mean Absolute Error (MAE) (Equation (1)) measures the mean of the absolute difference between the predicted time and the actual time. This measure is easy to understand because the unit is the same as the prediction. Therefore, the value of the MAE represents the average minutes that a prediction is above or below the actual time. It is used by Corman & Kecman (2018) to compare the performance of different BN models of the Swedish railways and Wen et al. (2019) to compare a Random Forest and a Neural Network.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \bar{y}| \quad (1)$$

Mean Square Error (MSE) (Equation (2) is the average of the squared difference between the predicted time and the actual time. Because the differences are squared, outliers are penalized more with MSE. Taking the root of the MSE results in the Root Mean Square Error (RMSE) (Equation (3). This is often used because the order of the unit is the same as the predicted unit, for the predictions this is minutes. Lessan et al. (2019) used the RMSE, MAE and mean error to quantify the performance of BN models for the Chinese railway.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

R^2 is the coefficient of determination (Equation (4). It describes how much of the total variation in the response variable can be explained by the input variables. R^2 normally varies between 0 and 1. When the value is close to 1, almost all the variability in the output variable is explained by the input variables. When the value is close to 0, the variables do not explain the variance in the output variable (Hastie et al., 2008). R^2 is used by Grandhi (2019) to compare the performance of models due to the benchmarking effect that the R^2 value offers.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{4}$$

3.7.2 Prediction quantiles

For regression, the conditional mean is given as the prediction. To determine the certainty or the spread of predictions, *Quantile Regression* (Koenker, 2005) can be used to predict conditional quantiles by adapting the loss function to a quantile loss function displayed in Equation (5) with α as quantile. For the median (50th quantile), the quantile loss function is equal to the sum of absolute errors. Errors above and below the predicted value are penalized evenly. For the 10th quantile, the loss of underestimation is less than overestimation. Figure 18 shows the quantile loss for different quantiles.

$$L_\alpha(y_i, f(x_i)) = \begin{cases} \alpha|y_i - f(x_i)| & \text{if } y_i > f(x_i) \\ (\alpha - 1)|y_i - f(x_i)| & \text{if } y_i \leq f(x_i) \end{cases} \tag{5}$$

Where y_i is the true value and $f(x_i)$ is the predicted value for data point i

Figure 18. Quantile loss per error

With *Quantile Regression Forest* (Meinshausen, 2006), the full conditional distribution can be determined with a Random Forest by storing all observations in the leaves of the trees instead of only the mean. This makes it possible to use one model to determine multiple quantiles. Gradient Boosted trees use the loss function during training. Therefore, multiple models have to be trained to determine multiple quantiles.

3.7.3 Feature importance

Permutation importance represents the decrease in performance of the model when a single feature is shuffled randomly. The value shows how important a single feature is in the model. Permutation importance is, instead of impurity-based feature importance, not bias toward high cardinality features and is not only determined on the training data and therefore represents also unseen data (Breiman, 2001).

3.7.4 Final model validation

Evaluating the final model on a selected test set can give high variance because it can depend on the split of the original dataset. For evaluating the performance of the model on unseen data, cross-validation can be used. Cross-validation tests the model with a subset of the data and trains the model on other k-1 subsets. The results from cross-validation are not dependent on the split in training and test set and give therefore a good representation of the end performance of the model on unseen data.

3.8 Summary

In this chapter, the previous research at ProRail into prognosis was described. The use of prediction intervals of probability distributions by de Wit (2016) showed good performance. The Bayesian Network of Zilko (2017) proved to be a good representation of an incident but had low prediction performance. DataLab ProRail (2019) showed that identifying the cause can improve the prediction reliability and that the later an updated prognosis is given, the more reliable it is. The research of Wemelsfelder (2019) combined features of previous research into models for an updated prognosis. These models showed minimal improvement compared to the initial prognosis and minimal impact of overprediction on the prediction errors. In other international prediction studies for railway, machine learning methods showed good performance. Therefore, the steps of the supervised machine learning are shown, and different steps are explained. First data pre-processing with cleaning and handling missing data is explained. Secondly, methods for feature selection are described. Also modelling methods linear regression, tree-based methods, bayesian networks, artificial neural networks and support vectors machines and their results in previous projects are described. To improve the performance of these methods, grid search and random search are described as methods for hyperparameter tuning. For model evaluation, the MAE, MSE, RMSE and R^2 metric are identified to measure the ability of the models to predict for new data. Besides point predictions itself, also the prediction uncertainty and Quantile Regression with the quantile loss function is identified to communicate prediction percentiles. Permutation importance is described as a method to determine the features with the biggest impact on the prediction performance of a model. Finally, cross-validation is described as the method for validation of the final model.

4 Data pre-processing

In section 2.4.3 two incident types were marked by ProRail as the most interesting incidents for this thesis because they occur often and do not involve a third party: ‘defect material’ and ‘collision with a person’. In Section 4.1 is explained why ‘collision with a person’ is selected to develop a prediction model. In Section 4.2 the collection of raw data for model creation is described. In Section 4.3 is described how this data is cleaned and filtered. In Section 4.4 is explored which features in the data will be used for the prediction model by feature selection. In Section 4.5 transformations to the data are described to make it ready for the models.

4.1 Incident type selection

Based on the analysis of the reliability of the final prognosis for the incident types defect material and collision with a person in Section 2.4, an incident type will be selected for which a prediction model will be developed. The selection for the incident type is made based both on the current reliability and the impact that an improvement in prognosis reliability that the selected incident type can have on the train operations.

Defect material is an incident type that has high uncertainty. Defect material refers to trains that do not drive. To resolve this, first, the train operator has to try with a helpline for 25 minutes to restart the train itself. If this does not work, the train has to be removed with the help of another train. This can take a long time and it is mostly dependent on the availability of trains in the area of the defect train.

For an incident collision with a person, the impact on the train traffic is high and the recovery process is clear. The initial prognosis for this type of incident is 180 minutes and all the tracks at the incident location cannot be used. Because these incidents are long and have a mostly fixed process, there is enough time for a prognosis. During the process, multiple parties are involved such as government emergency services, the incident recovery team and the train operator. The times of these parties are logged during the incident recovery process, so there is more data available compared to the incident type defect material. From the first plot in Section 2.4.3 of collision with a person, it can be seen that half of the prognoses are not *in time*. Although, the prognoses are *precise*, a reduction in overprediction would result in less additional waiting time. Due to the amount of data that is available during the incident recovery process and the impact on the train operations, ‘collision with a person’ is a good incident type to determine whether a model can improve the reliability of the prognosis of the end time.

4.2 Data collection

Data about incidents is stored in the SpoorWeb application, which was introduced by ProRail in June 2017. For model creation, all incident data from June 2017 to mid-November 2020 is used. This data consists of three parts:

1. General incident information with prognoses
2. Tasks per party involved in the incident, with their arrival times and prognosed task duration.
3. Logs of the incidents. These logs are written in textual format and contain the data of the general incident information and the task per party.

Next to SpoorWeb, data on weather conditions at the time of the incidents is retrieved from KNMI. This data contains the average hourly temperature in De Bilt in The Netherlands and information on rain, snow, mist, thunder or icing in the hour in which the incident occurred.

In total, 807 collision with a person incidents are logged in SpoorWeb between June 2017 and November 2020. Incidents without an end time are removed, after which 790 incidents remain.

For the 'collision with a person' incidents, data for different activities during the incident is stored. The general incident information contains 92 features with description about the location of the incident, the reporter and the time of occurrence and prognoses. Also, the actions for the train traffic are included. The tasks per party data consists of 425 features with information per party. For example, if a party goes to the incident location, their estimated arrival time and the actions performed with their expected duration is stored.

The first selection of features, after discussion with general leaders and a data specialist of the incident process at ProRail, resulted in 29 features from the general information. For the task data, features with more than 20% missing (with exception of the features for "Schouwarts") and categorical features with one category were removed. After removing data that was already present in the general data, 51 features of the task data remained. An overview of all general, task and KNMI features with a description, number of categories, and percentage of missing data can be found in Appendix A.

4.3 Data pre-processing

Before the most important features can be selected, data has to be cleaned by removing outliers and features with missing data and some features have to be modified. Then data with outliers have to be removed and missing data has to be filled in.

4.3.1 Feature modification

The time features are transformed into the minutes past the time the incident occurred. These numerical features are easier to use in the model and can later be transformed back to a time. The moment in which an incident occurred, is transformed into the hour, day of the month, day of the week and week of the year. Also, the binary features night and rush hour are introduced because these were identified by de Wit (2016) and Zilko (2017) as important distinctions. When an incident occurred between 00:00 and 6:00, Night is 1 and otherwise 0. Rush hour is 1 between 6:30 and 9:00 and 16:00 till 18:30 and otherwise 0.

For the logistics and diversion regulations, the information can be changed during the incident when new information becomes available. The last value selected was identified by the logistics experts as best representative for the incident.

4.3.2 Outliers

A common method for outlier detection is the Z-score, which excludes points more than 3 standard deviations from the mean. This is, however, under the assumptions that the data is normally distributed. Because the data on selected features may not be normally distributed, the outlier cut off is set based on analysis of the features. Because time features are transformed into minutes past the time the incident occurred, negative time values are not allowed. Therefore, incidents for which time features are negative are removed. The average duration of a collision with a person incident is around 3 hours, incidents with time features of more than 1000 minutes (16.6 hours) an also excluded. After these selections on time features, 717 incidents remained.

4.3.3 Missing data

For many incidents, data is missing on specific features. There can be two reasons. It did not apply to the incident, for example, the police did not have to go to the incident location. Or the data was not entered in SpoorWeb.

Some prediction models, such as, tree-based methods can handle missing data, but others require the missing data to be filled. As described in Section 3.2 there are multiple strategies for filling missing data. For this project, all missing data will be imputed to ensure similar data for all modelling methods.

When the initial prognosis is missing, it is imputed with a constant value of 180, which is the value given by the current CQM decision tree. For the number of changes to the train schedule, mutations, diversions, actions to perform according to diversion regulation, missing values are filled with a 0 because it was identified that if this data was missing, no actions were performed. For incidents where data for tasks is missing, it is not missing at random. The data is missing because the task was not performed or not filled in. The time features of these tasks are therefore imputed with 0, which resembles that the duration of this task was 0 for this incident.

For the categorical features where data is missing, an extra category of “Unknown” is created. This category is used when the data is missing at random, as well as for data that is not missing at random.

4.4 Feature selection

The filter, wrapper and embedded methods described in Section 3.4 can be used for feature selection, Filter methods are used to gain insight into the data. Wrapper methods can be used for models that have no embedded feature selection. From the embedded methods LASSO regression and tree-based methods are used.

4.4.1 Filter methods

The filter methods will be used to gain insight into the data and identify the important features.

Numerical features

Correlation can be used to show the relationship between numerical features. A Pearson Correlation is performed to calculate the correlation between all numerical features. The full correlation matrix can be found in Appendix A. Figure 19 shows the correlation of the numerical features with the incident duration (Einde Incident).

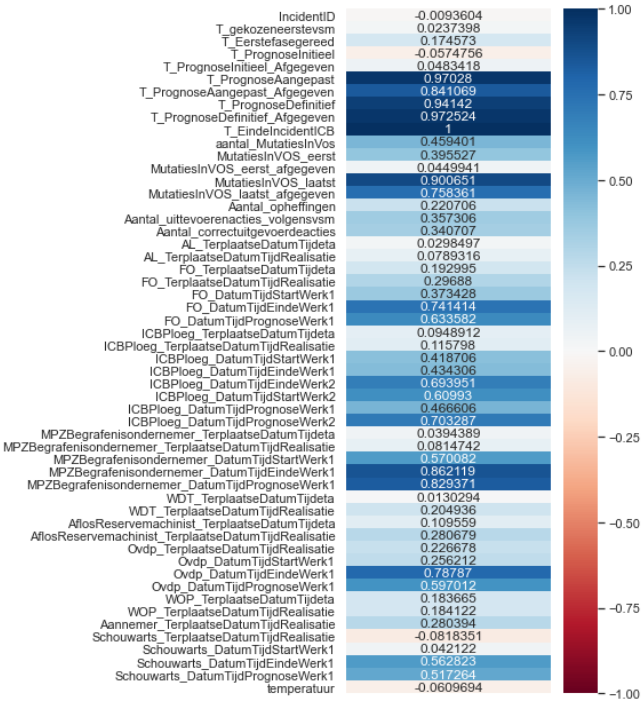


Figure 19. Correlation with Incident duration

The correlation with incident duration in Figure 19 shows high correlation for actions that are done at the end of the incident, for example, the moment in which an updated or final prognosis is given. Other important time features are the end of the forensic investigation team activity, the end of the work of the mortician and the end of the second work activity of the incident recovery team. From the features that are known at the beginning of the incident, the number of actions according to the diversion regulations and the start of the ICB team show small correlation with incident duration.

Categorical features

For the categorical features, a correlation is not possible. Therefore, an ANOVA is used to test the difference between the means of the incident duration for the levels of each categorical feature. An example of the difference in the mean of the incident duration of the degree of fragmentation feature is given in Figure 20. The *degree of fragmentation* is the amount of the fragmentation of the victim of the collision with a person incident.

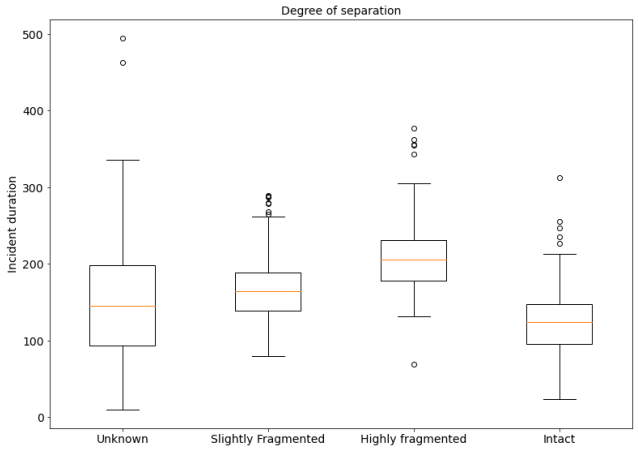


Figure 20. Degree of fragmentation boxplot

The ANOVA values for the categorical features with the incident duration in Figure 21 show that for a significance level of 0.05, only the hour of the incident time has a significant impact on the incident duration.

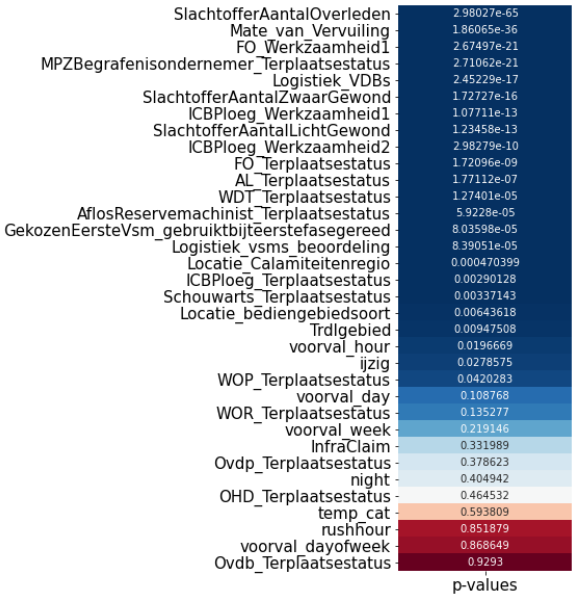


Figure 21. p-values ANOVA with incident duration

4.4.2 Wrapper methods

For methods that do not include feature selection, a wrapper method can be used. A wrapper uses the feature coefficients or feature importance from other models and wraps around it to select the best features.

As a wrapper method, the Recursive Feature Elimination (RFE) is used around Extremely randomized trees (Extra Trees). Extra trees differ from Random Forest in the way a split is made. With Random Forest, the best split is selected and with Extra trees, this split is random, which makes them computationally faster (Geurts et al., 2006). RFE eliminates features until the performance metric, in this case, R^2 does not increase. This process is performed with cross-validation to reduce bias.

RFE selected 21 features of which most have a high correlation with the incident duration or high p-values in the ANOVA. However, in the RFE also features were selected that had low correlation values. The full list with features from the RFE can be found in Appendix B.

4.4.3 Embedded methods

Embedded methods have the features selection built in. Therefore, feature selection is not necessary before using these methods. Lasso is used for linear regression, which, as described in Section 3.4, reduces the weights of features that have minimal impact on the prediction performance.

Random Forest performs feature selection based on the way splits in the trees are made. The features with higher importance are mostly used for splits and less influential features will not be used for splits.

4.5 Data preparation

Before the features from the feature selection can be used in the models, some transformations have to be performed.

The categorical features have to be transformed into a numerical representation. For this One-Hot-Encoding can be used, which creates a column for every category in a categorical feature with a binary variable. For the column of the first category, only the rows that contain the first category will be “1” and the rows with other categories “0”.

The numerical features have to be transformed for the models to handle the differences in the magnitude of the numerical features. For example, the time in minutes is of a different magnitude than the number of changes in the train schedule. When features are not of the same magnitude, one might dominate in the objective function. For this, min-max normalization is used to scale the values of all numerical features to the range of [0,1].

4.6 Summary

‘Collision with a person’ is chosen as the incident type to create a prediction model because this incident type has a big impact on the train operations and more data during the incident recovery process is available. For the model, data between June 2017 to November 2020 from SpoorWeb and KNMI is used. The data is cleaned and incidents with time features that are negative or larger than 1000 minutes are removed. In the pre-processing, features are modified, outliers are removed, and missing data is imputed. For the feature selection, first filter methods correlation for numerical features and ANOVA for categorical features are used to gain insight into the data and identify important features. A wrapper method is used to perform feature selection for methods that do not have embedded feature selection. Finally, the features are normalized to have the same magnitude. In the next chapter, the setup for model comparison and moments of prediction is explained.

5 Model development and results

In chapter 4, the raw data is cleaned, features are selected by feature selection techniques, and data pre-processing steps are explained. In this chapter, a machine learning method is selected and used to build a model for incident end time prediction.

The goal of this project is to determine if a data-based prediction model can support more reliable prognosis for the incident end time when new data becomes available during the incident. To select the best method for prediction, the prediction performances of the 7 machine learning methods described in Section 3.5 are determined in Section 5.1 and the best method for prediction is selected. The selected method is used in Section 5.2 to create a model for prediction of the end of incident time at chronological stages during the incident. In Section 5.3, the model is used for three new incidents, to generate a new end time prediction whenever new data becomes available during the incident. Section 5.4 describes how prognoses can be determined with the incident end time predictions.

5.1 Methods for prediction

From the literature search 7 methods, Linear Regression, a Decision Tree, Random Forest, Gradient Boosted Trees, Bayesian Network, Neural Network and Support Vector Machine, were identified for prediction of the end time of incidents.

To determine which of these methods has the highest prediction performance, each method is tested using all data that is available at the end of the incident. Based on the prediction performance, the best method is selected. This method will be used to make predictions during the incidents when new data becomes available.

For a Bayesian Network, the joint distribution of the incident duration and the activities during the incident is difficult to construct. Because of this, the development of a Bayesian Network would take too much time for this project. Therefore, the Bayesian Network method is excluded. The six other methods are implemented using Python with the Scikit-learn package (Pedregosa et al., 2011), XGBoost package (Chen & Guestrin, 2016) and Tensorflow (Abadi et al., 2015).

5.1.1 Test setup for selecting the best prediction method

Before determining the performance of each method, the data pre-processing steps as described in 4.3 are performed. To improve the model performance, the hyperparameters are tuned with the Random Search method. The hyperparameter search space per method is described in Appendix C.

The performance of each method is determined by 5-fold cross-validation of the entire dataset, called the outer cross-validation. In each of the five runs, four training folds of outer cross-validation are selected to tune the hyperparameters. With Random Search, 25 random combinations from the hyperparameter search space are selected. The performance of every combination is measured with another cross-validation of 4 folds, the inner cross-validation (See Figure 22). The best combination of hyperparameters is used to measure the performance on the test set of the outer cross-validation. The metric used to measure the performance of the hyperparameter combination is R^2. In total 5 (outer folds) * 25 (random combinations) * 4 (inner folds) = 500 fits are performed.

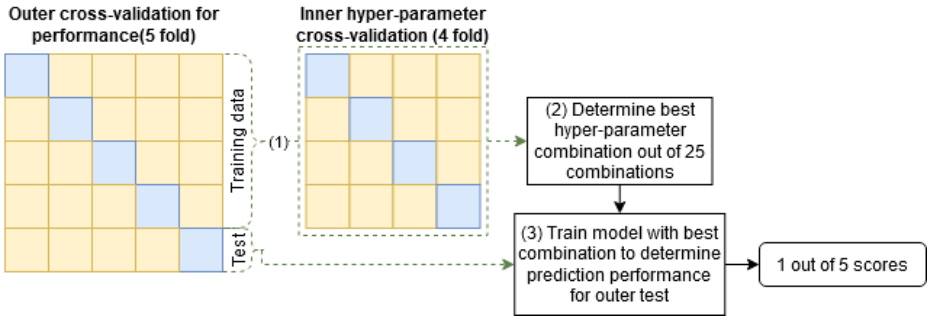


Figure 22. Nested Cross-validation

5.1.2 Performance prediction methods on all data

The prediction of the method with tuned hyperparameters for the 5 outer test folds results in 5 test performances. The average of the 5 test performances is used to compare the methods. For the outer cross-validation R^2, RMSE and MAD (Section 3.7.1) are determined as performance metrics. The results are displayed in Figure 23.

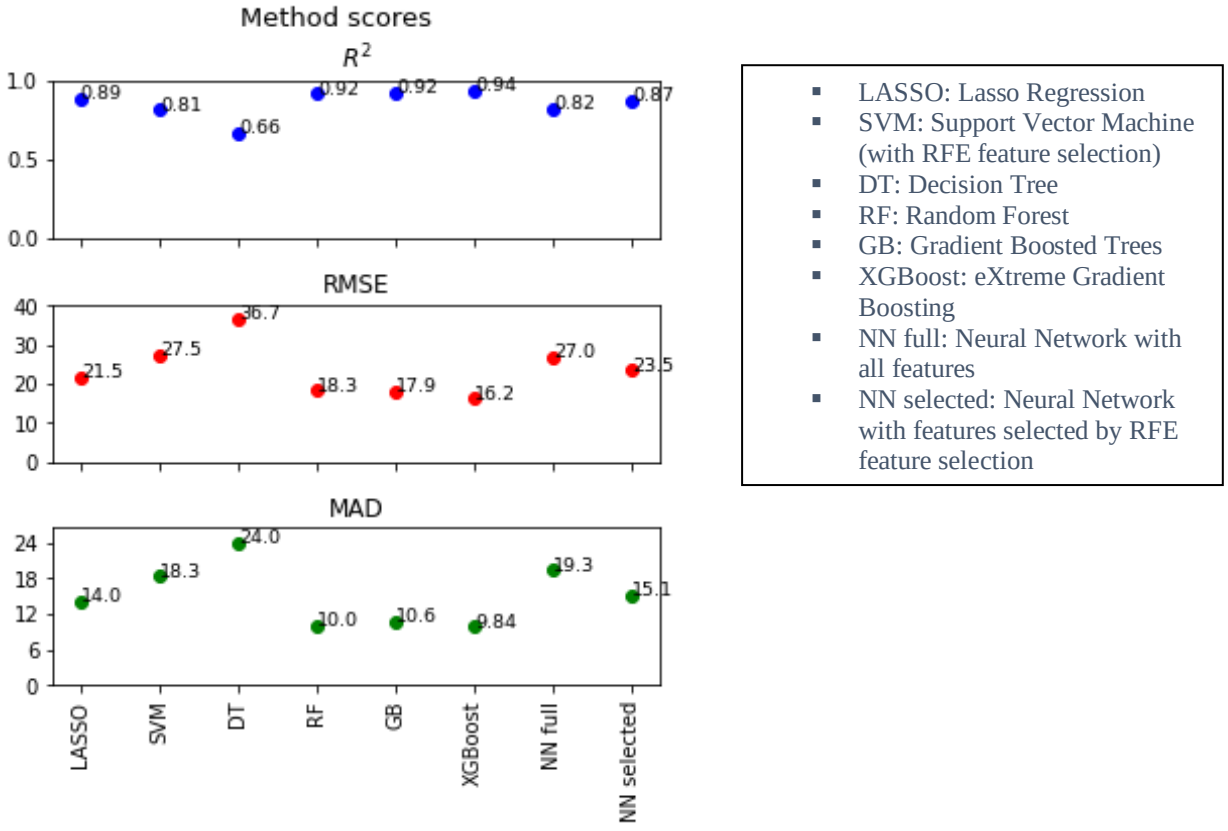


Figure 23. Performance of the modelling methods

The Decision Tree (DT) showed the lowest performance, followed by the Support Vector Machine (SVM), the Neural Networks with and without feature selection and Lasso Regression (LASSO). Random Forest and Gradient Boosted Trees (GB) show good performance, but the best prediction performance is achieved by eXtreme Gradient Boosting (XGBoost) with a mean absolute error for prediction of the incident end time of less than 10 minutes. Therefore, XGBoost will be used to make predictions for the end of incident time during the incidents in Sections 5.2 and 5.3.

For the decision tree, the training performance is higher than the test performance which indicates that the decision tree overfitted resulting in low test performance. The Neural Network showed different performance between the cross validation folds indicating that the network did not learned the patterns in the data from the minimal available number of incidents. The performance of the SVM and Lasso showed that part of the variance in the output variable could be explained by fitting hyperplanes and shrinking coefficients. The performance of the Random Forest is the result of the multiple random build decision trees. By averaging the results from the trees, the model reduces the variance while utilizing the good predictive performance of the individual trees. In the Gradient Boosted methods, the sequential build trees try to improve the points with low prediction performance. This helps the models to improve wrongly predicted points and therefore reduce the error.

5.1.3 Feature importance of XGBoost

Feature importance can give insight into which features have a big impact on the performance of the model. Tree ensemble methods have built-in impurity-based feature importance. However, for this project the feature importance is determined with permutation importance (Section 3.7.3). Figure 24 shows the permutation importance of the XGBoost model for the top 20 features.

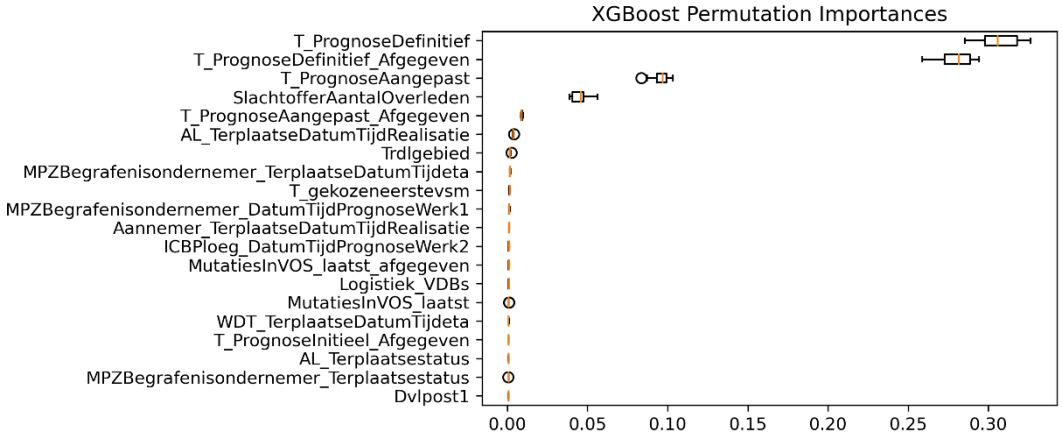


Figure 24. XGBoost Permutation feature importance

The most important features for the model are the final prognosis (T_PrognoseDefinitief) and the time in which the final prognosis is given (T_PrognoseDefinitief_Afgegeven). This is followed by the updated prognosis (T_PrognoseAangepast) and the number of victims deceased (SlachtofferAantalOverleden). The prognosis features are the predictions for the end of the incident time given by the AL during the incident. The high importance of these features shows that these features have a big impact on the ability of the model to predict the end time of the incident. This is expectable because these prognoses (and especially the final prognosis) are given when the AL is certain the incident will be finished by that time.

5.2 Prediction at incident stages

Each collision with a person incident is unique. Because of this, the parties that are involved in an incident and the order in which these parties arrive can be different. To evaluate the prediction performance when certain information is known, multiple chronological stages are defined by features that are selected in dialogue with several AL's. This enables comparison of the incidents and determination of the most important features for prediction. A full overview of all stages identified with their features is shown in Appendix D.

For example: at the begin stage, all features that are known when an incident is reported are included. And stage 1 includes the first information about the estimated arrival time of the AL and the ICB team.

5.2.1 Setup prediction stages

In Section 5.1.2, XGBoost was identified as the best method for predicting the end of incident time. Therefore, an XGBoost model is built to predict the end of incident time at each stage using the features that are available. The performance is determined with the same setup of cross-validation described in Section 5.1.1. To evaluate the certainty of the end of incident time predictions, prediction intervals are be determined at every stage by Gradient Boosted Trees with quantiles. In Section 5.2.2 the prediction performance at each stage is discussed followed by the important features for the model in Section 5.2.3. The prediction intervals per stage are explained in Section 5.2.4.

5.2.2 Prediction performance per stage

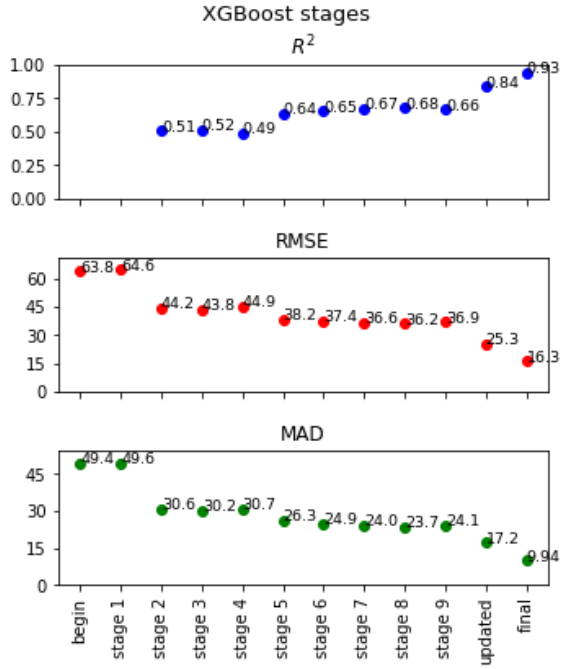


Figure 25. Prediction performance at different stages during the incidents

The performance of the model at different stages according to R^2, RMSE and MAD in Figure 25 shows the biggest performance improvement when the updated and final prognoses become known. Although the prognoses are important, improvements can also be seen in the stages during the incident. For the begin stage and stage 1 the performance is low but at stage 2 a clear improvement is seen. After this stage, the errors slowly decrease till stage 9 as more information about the incident becomes known.

5.2.3 Feature importance per stage

For each stage, the feature importance is determined with permutation importance (Section 3.7.3). The top 20 important features for the stage with the biggest performance improvement, stage 2, is shown in Figure 26. The feature importance for stage 9, when all data about the incident is available is shown in Figure 27. The important features at the final stage are the same as on all data (see 5.1.3).

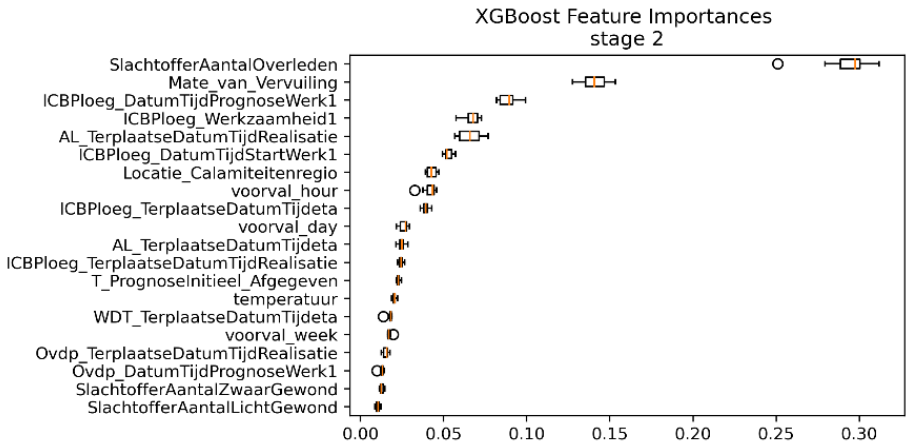


Figure 26. Feature importance stage 2

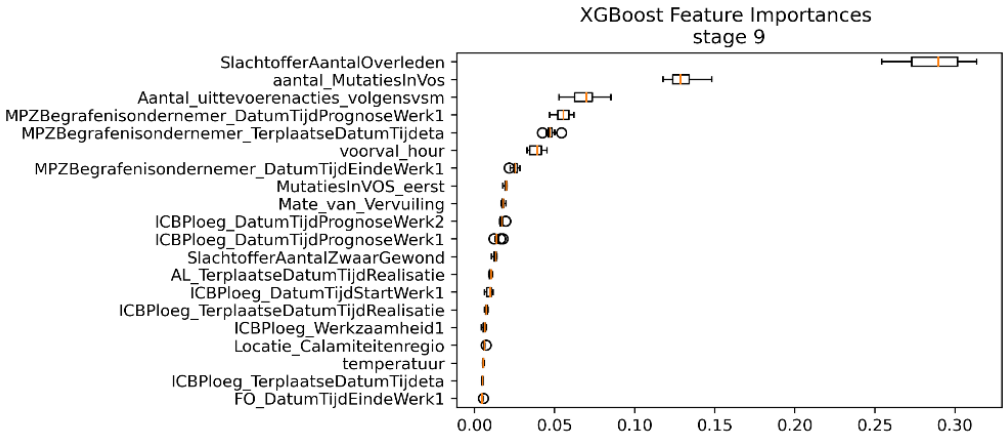


Figure 27. Feature importance stage 9

At stage 2, the most important features for the model are the number of victims deceased (SlachtofferAantalOverleden) and degree of fragmentation (Mate van Vervuiling). For stage 9 the number of victims deceased is still the most important feature, other important features are the estimated time of arrival (MPZBegrafenisondernemereta), end time of the working activity (MPZBegrafenisondernemerEindeWerk1) and prognosed finish time (MPZBegrafenisondernemerPrognoseWerk1) of the mortician.

5.2.4 Prediction intervals

Prediction intervals to communicate the uncertainty in predictions for each stage are determined with the Gradient Boosted Trees quantiles (described in Section 3.7.2). The 0.05 and 0.95 quantiles were used to determine the 90% prediction interval. The prediction intervals (PI) for the begin stage, stage 2 and final stage are shown in Figure 28, Figure 29 and Figure 30 respectively. In the figures, the red dots are the actual incident end times. On the x-axis, the incidents are ordered based on the width of the prediction interval. The y-axis shows the actual duration of the incidents.

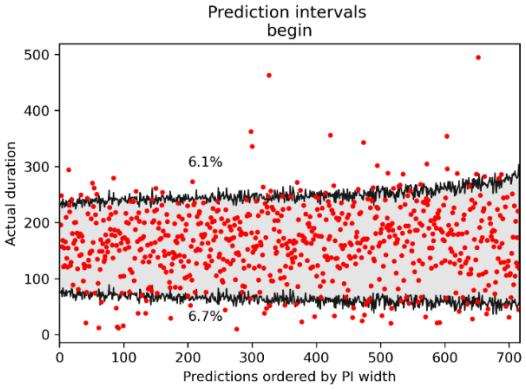


Figure 28. Prediction intervals stage begin

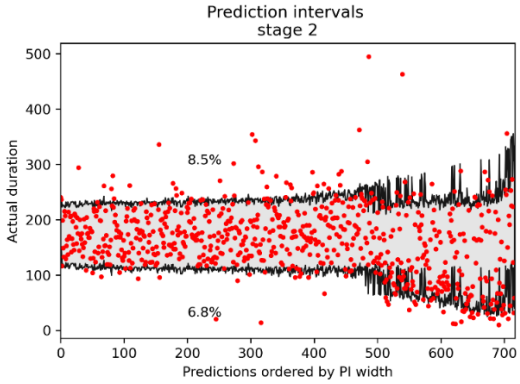


Figure 29. Prediction intervals stage 2

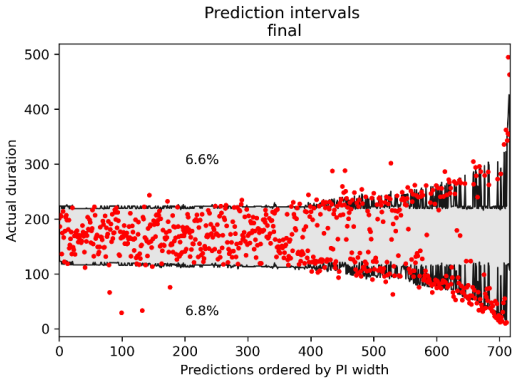


Figure 30. Prediction intervals stage final

The width of the PI's decreases for the incidents with the smallest prediction intervals from 150 minutes in the begin stage (Figure 28) to 100 minutes at the final stage (Figure 30). This shows that with more information, the variability in the predictions can be explained better. However, the prediction intervals are still wide with 100 minutes at the final stage meaning that the uncertainty in the predictions remains high.

At the begin stage (Figure 28), the incidents show an even spread of duration. In stage 2 (Figure 29), many incidents with bigger PI's, on the right side of the figure, appear to be incidents shorter than 100 minutes. At the final stage, a clear split can be seen in actual duration for incidents 450 to 717, which are the incidents with a big PI at the final stage (Figure 30). These figures show that the split between short and long incidents can be distinguished better when more information is available. A decision tree is used to identify which feature best explains the split in short and long incidents. From the decision tree, the number of victims (SlachtofferAantalOverleden) was identified as the feature that best explains the difference. Figure 31 shows that when 0 victims are deceased, incidents have an average duration of 90 minutes and when 1 or more victims are deceased the average duration is 200 minutes.

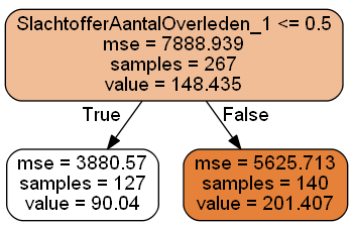


Figure 31. Decision Tree showing most important feature for split in short and long incidents. Value = duration

5.3 Prediction throughout incidents

In order to demonstrate the application of the XGBoost model for incident end time prediction during an incident, 3 new incidents are selected from the period of December 2020 to January 2021. Instead of using stages, the incident end time is predicted every moment new information becomes available for the new incidents. Using all features that are available at that moment, the XGB model generates a new prediction of the incident end time.

The CQM decision tree for the collision with a person incident type (Figure 32) includes the degree of fragmentation feature. Because this feature is not available during the intake of the incident, the root of the tree (180 minutes) is used for the initial prognosis. This CQM decision tree is only used during the intake and not used to update the prognosis when the degree of fragmentation becomes available.

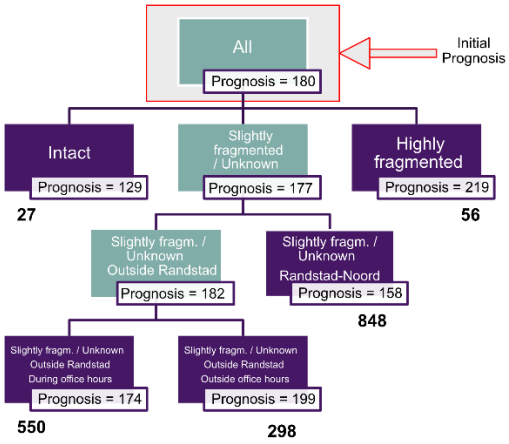


Figure 32. CQM decision tree collision with a person

5.3.1 Prediction results for three new incidents

The predictions for incident 1 are displayed in Figure 33, for incident 2 in Figure 34 and incident 3 in Figure 35. On the x-axis is the number of minutes since the start of the incident. On the y-axis is the predicted incident end time. For comparison with the initial, updated and final prognosis, the prognosis of the CQM model of Figure 32 is included when the degree of fragmentation becomes known. It is worth mentioning that the incident end time prediction focuses on the mean of the errors and therefore, it does not overpredict. For the prognoses, overestimation is desirable since underestimation leads delay because the restart plan has to be changed. In Section 5.4, a method will be explained regarding how to derive a prognosis from an incident end time prediction.

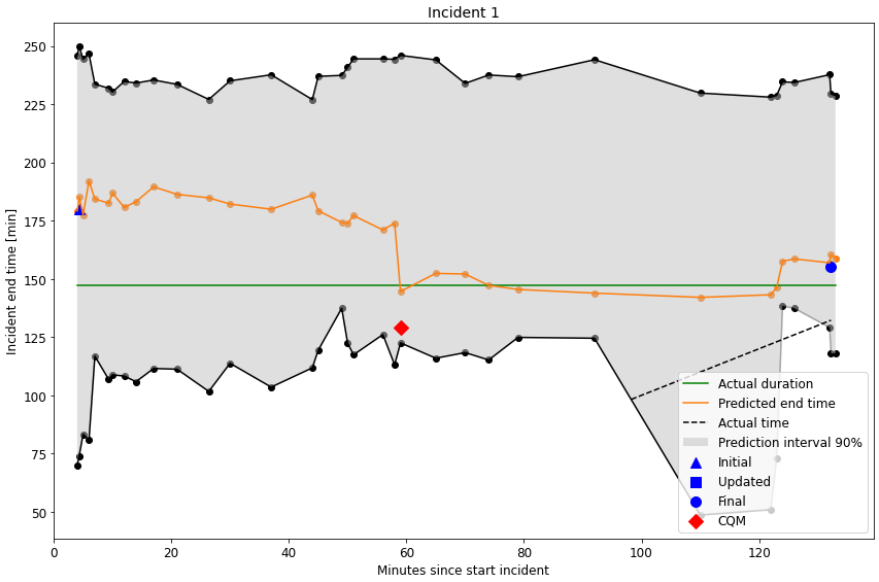


Figure 33. Timeline incident 1

For incident 1 (Figure 33), the predicted incident end time from the model (orange line) at the start of the incident is similar to the initial prognosis. Around 60 minutes into the incident, when the degree of fragmentation becomes known, the prediction changes to the actual duration of the incident. After 60 minutes, the prediction remains around 10 minutes from the actual end time. In this incident no updated prognosis was available and, therefore, the initial prognosis was used until the final prognosis right before the end of the incident.

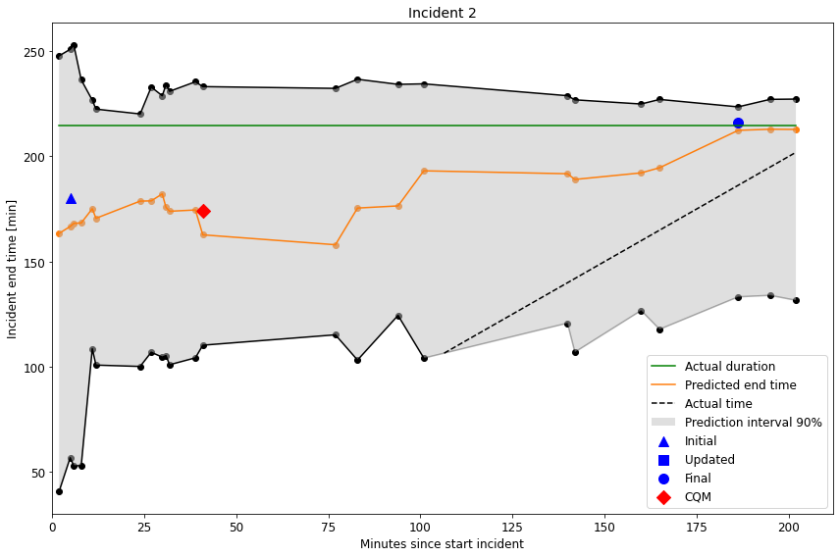


Figure 34. Timeline incident 2

For incident 2 (Figure 34), the predicted end time is during the whole incident below the actual duration. At minute 40 the degree of fragmentation is determined and the prediction decreases. When at minute 75 the mortician and the ICBPloeg arrive at the incident location and around 100 minutes the forensic investigation team arrives at the incident location, the prediction improves. At the time the final prognosis is given, the prediction end time remains around the actual incident end time. In this incident also no updated prognosis was available and therefore the initial prognosis was used until the final prognosis right before the end of the incident.

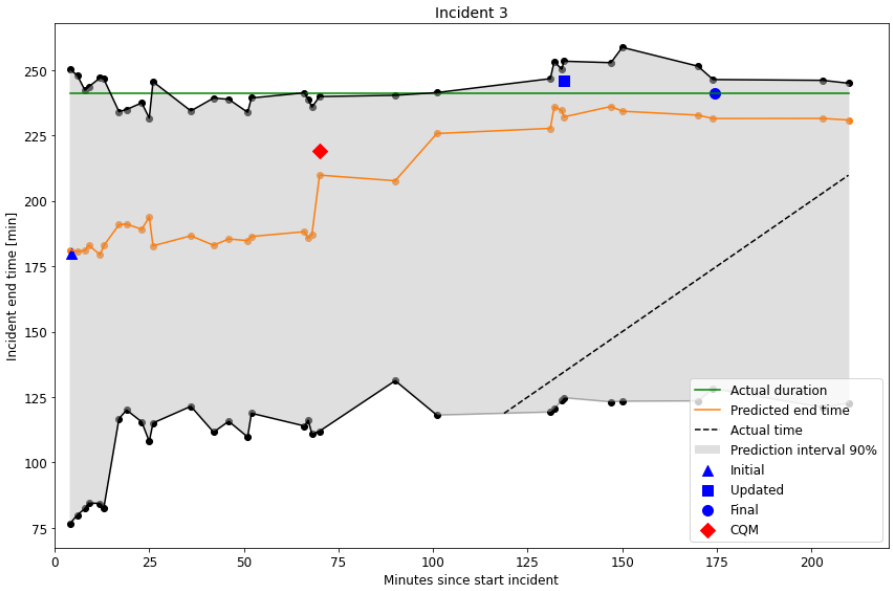


Figure 35. Timeline incident 3

Incident 3 (Figure 35), shows that the predicted incident end time starts 60 minutes below the actual incident duration. During the incident, the prediction improves and after 125 minutes the prediction remains within 15 minutes from the actual duration. The biggest improvement of the prediction was at 70 minutes when the degree of fragmentation was determined and at 90 minutes when the mortician arrived. The updated prognosis showed a clear improvement over the initial prognosis which was also included in the prediction of the model.

5.3.2 Observations

In the current situation only the initial, updated and final prognosis are given. With a prediction model that determines a new prediction of the incident end time every moment new data becomes available, a prediction can also be given between the times of the current prognoses. For the second half of the incident, the predictions of the model show an improvement over the initial prognosis. In both incident 1 and incident 3, the prognosis improved when the degree of fragmentation was determined. The prediction of the CQM decision tree using this feature shows for both incidents a more accurate prediction compared to the initial prognosis. The width of the prediction intervals is, for all incidents, around 200 minutes at the beginning of the incident. This decreases to about 120 minutes during the first half of the incident. In the second half of the incident, the prediction interval is bounded by the number of minutes since the start of the incident. Because of this, the width of the interval decreases for longer incidents.

5.4 Incident end time prediction to support reliable final prognoses

The XGB model can determine a new prediction for the incident end time when new data becomes available. Because of this, predictions are made throughout the incident recovery process. The last known prediction is based on all information available at that time and therefore, it provides the best possible prediction of the incident end time. These predictions can give insight into the development of the incident duration and support the AL to give *in time* prognoses.

For prognoses, underprediction is less desirable than overprediction. Therefore, the 65th percentile of the distribution is used in the CQM decision tree for the initial prognosis. At the end of the incident, underprediction leads to delay. The distribution of the residuals of the final prognosis in Figure 36 shows that 90% of the final prognoses are *precise*. Meaning they are an overprediction of the actual duration.

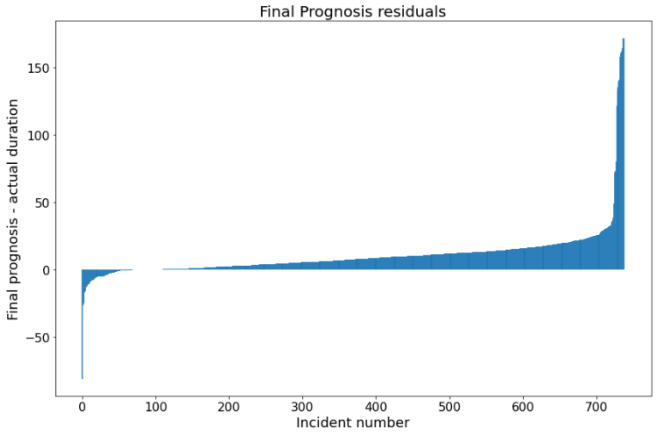


Figure 36. Final prognosis residuals

Prognoses can also directly be determined from the prediction of the XGBoost model, which penalizes under and overprediction equally, by adding several minutes to the prediction to achieve the desired percentage of overprediction. This number of minutes can be determined from the distribution of the residuals between the actual incident duration and the predicted incident end time of the XGBoost model. Figure 37 shows that for the XGB model with all data, 12 minutes have to be added to the incident end time prediction at the final stage to have overprediction for 90% of the incidents, as currently achieved with the final prognosis.

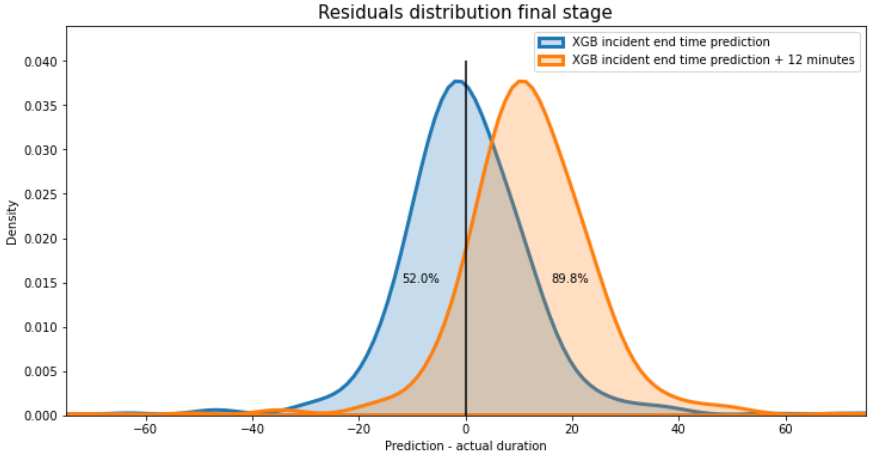


Figure 37. Residuals incident end time prediction with and without overprediction

6 Conclusion

In the current process, a decision tree developed by CQM is used to determine a prognosis at the intake of an incident and prognoses during the incidents are only based on the expertise of the AL. No data-based methods are used to update the prognoses during the incident recovery process. Previous research at ProRail on reliable prediction for the incident end time showed that incident duration is difficult to predict and the uncertainty about the predictions is high.

Data analysis in Section 2.4.1, shows that the final prognoses, given by the AL, are for the most occurring incident types in 2020 not given *in time* (35 minutes before the end of an incident). The final prognoses are *precise* (prognosed incident end time after actual end time), but show overprediction for most incident types, leading to an additional waiting time before the restart of the train. For this project, the incident type collision with a person is selected because enough historical data is available and more than half of the final prognoses are not *in time*. Collision with a person final prognoses are *precise*, although they show overprediction.

The goal of this project was to create a data-based model for incident end time prediction to support *in time* and *precise* prognoses. International research in the railway industry shows the use of machine learning methods for incident duration and delay predictions. From the literature in 3.5, 7 methods are identified for incident end time prediction in railways. In Section 5.1, 6 of these 7 methods are applied to all incident data of the collision with a person incident type. From these methods, the XGBoost model has the highest prediction performance for predicting the incident end time. From the feature importance of the XGBoost model, the updated and final prognosis are found to be important features to predict the incident end time at the end of the incident. These features also show high correlation with the incident end time in Section 4.4.1. This means that none of the other features, collected during the incident recovery process, can support a better prediction than the updated and final prognosis.

For the prediction of the incident end time during an incident, the XGBoost model is used in Section 5.2 to determine new predictions when new data becomes available at different stages during the incidents. The model shows that the prediction performance improves when more information becomes available at these different stages throughout the incident. At stage 2, when the number of deceased victims and the degree of fragmentation of the body of the victim(s) becomes known, the prediction performance shows a big improvement. These categorical features also show very low p-values for the ANOVA test in 4.4.1 and, therefore, a significant difference in the means of the categories. The degree of fragmentation was already identified in the CQM decision tree for initial prognosis but because this feature is not available at the intake, the model is not implemented for this incident type. Although the prognoses of the AL show the biggest improvement in prediction performance, the stages show that from stage 2 it is possible to improve the prediction of the incident end time earlier in the incident process with the data that becomes available at these stages before the prognoses are known.

The prediction intervals for the stages during the incident in Section 5.2, show that the uncertainty about the predictions remains high throughout the incident. The 90% prediction intervals for the begin stage have a width of 150 minutes, which decreases to 100 minutes when all incident data is available at the final stage. Incidents with an actual duration of ≤ 100 minutes or ≥ 200 minutes show a wider prediction interval from stage 2. The difference between these short and long incidents is best explained by the feature number of deceased victims. This shows, once more, the importance of this feature for the prediction of the incident end time.

The application of the XGBoost model in Section 5.3 demonstrates the use for incident end time prediction every time new incident data is available. The model showed for 3 new incidents, that in the second half of the incident the prediction converges to the real incident end time. In 2 of the 3 new incidents the features that showed the biggest improvement in prediction are the degree of fragmentation, the arrival times of the mortician and the forensic investigation team.

The XGBoost model predicts the incident end time with even under- and overprediction and prognoses focus on overprediction. Therefore, the predictions from the XGBoost model cannot directly be compared with the current prognoses. However, prognoses can be determined from predictions by shifting the distribution of the residuals of the predictions to obtain *precise* prognoses with the desired percentage of overprediction.

This study shows that the developed XGBoost model for incident end time prediction during collision with a person incidents can be used to determine features that are important for prediction. The model can also be used during new incidents to give incident end time predictions when new data becomes available, as shown in Section 5.3. These predictions can give the AL insight into the development of the incident duration during the incident to support *in time* prognoses.

7 Discussion and recommendations

In this chapter, the used methods and found results are discussed, followed by recommendations for implementation and future research at ProRail.

7.1 Discussion

During an incident, information can be updated. Every time the information is updated, the changes are logged in log files and the data in the database is overwritten. A limitation of this project is that the data used for model development is downloaded from the database and, therefore, it only includes the last known information. This influences the performance of the models and feature importance.

When data is not missing at random, it contains important information about the incident process. For example, when an activity is not performed, the data is not missing at random but on purpose. As described in Section 4.3.3, for this project, data that is not missing at random and data that is missing at random is filled with the same value because it cannot be identified which one it is. Because of this, information about whether an activity is performed or not is lost.

The prediction performance of the methods is determined in Section 5.1 with cross-validation on the full data set. The prediction performance of the methods may be different when fewer features are selected, for example with only the data available at the stages. This means that another method than XGBoost might perform better for determining predictions during the incident.

As described in Section 5.1.1, only 25 random combinations from the hyperparameter search are evaluated for each method. When more combinations are evaluated, the prediction performance of methods could change, leading to another selected method for prediction.

The feature importance in Sections 5.1.3 and 5.2.3 shows the dependency of the model on each feature. Features that seem important in an early-stage model, may appear unimportant in a model at the end of the incident and vice versa. Because a model at an early stage is less robust than at a later stage, the importance of the features at stage 2 described in section 5.2.3 is less reliable than the importance of the features described for the final stage.

Lastly, the XGBoost model predicts the incident end time by reducing the mean squared error. This results in an even percentage of under- and overprediction. To determine a prognosis with a certain percentage of overprediction, as desired by ProRail, the predicted incident end time is increased. In the earlier stages, the uncertainty in the predicted incident end time is higher. Therefore, the predicted incident end time has to be increased more, resulting in long additional waiting times.

7.2 Recommendations

In this section recommendations and directions for future research at ProRail are described.

This project shows that for the incident type collision with a person it is possible to generate more *in time* prognoses using a data-based model for incident end time prediction. Researching if a data-based model can also have benefits for other incident types is recommended.

Based on the feature importance of stage 2 and the examples of the 3 new incidents, implementing the CQM decision tree which uses the degree of fragmentation for the collision with a person incident type is recommended. Another feature that could be investigated to extend the CQM decision tree is the number of victims deceased, which showed a clear difference between short and long incidents. Also, the arrival time of the AL, the estimated time of arrival and the prognosed end time of the work of the mortician could be interesting features as seen in the feature importance at stage 9 and for the full data. Further analysis of predictions from the XGBoost model during new incidents can provide more insight into the important moments and features with which the CQM decision tree can be extended.

A data-based model like the CQM decision tree can be used to determine more intermediate prognoses when data about identified features becomes available. This can be the first step towards a data-based decision support tool for the AL to update the prognoses every time new information becomes available.

The feature importance on all data shows that the updated and final prognosis are the most important features. Therefore, it is recommended to ProRail to keep these prognoses given by the AL, while determining extra moments for data supported prognoses.

For this project, a prognosis is defined as *precise* when the predicted incident end time is after the actual duration of the incident. High overprediction results however also in additional waiting time before the trains can be restarted. By specifying the penalty for different levels of under- and overprediction, it is possible to define a custom objective function, which can be implemented in the XGBoost model.

In Figure 29, a split between short and long incident with wider prediction intervals becomes visible at stage 2. Future research can explain why shorter incidents have larger prediction intervals and determine if the prediction intervals can be used for decision making.

The Bayesian network was excluded in this project because the dependency between the incident duration and the activities during the incident with a joint distribution is difficult to construct. However, as also identified by Zilko (2017), a Bayesian Network can be used to model the conditional distribution of the incident duration, which can be updated when new data becomes available. Therefore, future research into the construction of the conditional distribution with a probabilistic graphical network like Markov Network or Bayesian Network can be interesting.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur, M., … Zheng, X. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

Abu-Mostafa, Y. S., Magdon-Ismail, M., & Lin, H.-T. (2012). *Learning From Data*. AMLBook.

Akinsola, J. E. T. (2017). Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology (IJCTT)*, *48*, 128–138. https://doi.org/10.14445/22312803/IJCTT-V48P126

Allison, P. D. (2001). *Missing Data (Quantitative Applications in the Social Sciences)*.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*, 281–305.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-Augu*, 785–794. https://doi.org/10.1145/2939672.2939785

Corman, F., & Kecman, P. (2018). Stochastic prediction of train delays in real-time using Bayesian networks. *Transportation Research Part C: Emerging Technologies*, *95*(October 2017), 599–615. https://doi.org/10.1016/j.trc.2018.08.003

CQM. (2019). *Prognose herstelduur – resultaten analyses* (Issue april).

DataLab ProRail. (2019). *Section malfunction function recovery time prediction*. https://www.xomnia.com/post/helping-prorail-get-delay-predictions-on-track-with-machine-learning/

de Wit, N. (2016). *Development of a reliable prediction method for urgent infra-failure recovery times at ProRail B.V.*

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Ghaemi, N., Cats, O., & Goverde, R. M. P. (2017). Railway disruption management challenges and possible solution directions. *Public Transport*, *9*(1–2), 343–364. https://doi.org/10.1007/s12469-017-0157-z

Ghofrani, F., He, Q., Goverde, R. M. P., & Liu, X. (2018). Recent applications of big data analytics in railway transportation systems: A survey. *Transportation Research Part C: Emerging Technologies*, *90*(September 2017), 226–246. https://doi.org/10.1016/j.trc.2018.03.010

Grandhi, B. S. (2019). *Predictive Modelling using Machine Learning Techniques for Railway incident based parameters*.

Guyon, I., & Elisseeff, A. (2003). An Introduction to Variable and Feature Selection Isabelle. *Journal of Machine Learning Research*, *3*, 1157–1182. https://doi.org/10.1016/j.aca.2011.07.027

Hastie, T., Tibshirani, R., & Friedman, J. (2008). *Elements of Statistical Learning* (Issue 6). https://doi.org/10.1007/978-0-387-84858-7

Heerkens, H., & Winden, A. Van. (2017). Solving Managerial Problems Systematically. In *Solving Managerial Problems Systematically*.

Huang, P., Lessan, J., Wen, C., Peng, Q., Fu, L., Li, L., & Xu, X. (2020). A Bayesian network model to predict the effects of interruptions on train operations. *Transportation Research Part C: Emerging Technologies*, *114*(August 2019), 338–358. https://doi.org/10.1016/j.trc.2020.02.021

Jambhorkar, S. D. S., & Jondhale, M. V. S. (2015). *Data Mining Technique: Fundamental Concept and Statistical Analysis*.

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press. https://doi.org/10.1017/CBO9780511754098

Kooiman, N., de Jong, A., Huisman, C., van Duin, C., & Stoeldraijer, L. (2016). PBL/CBS Regionale bevolkings- en huishoudensprognose 2016–2040. In *Statistics Netherlands: Bevolkingstrends, september 2016* (Vol. 1, Issue september).

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. https://doi.org/10.1007/978-1-4614-6849-3

Lessan, J., Fu, L., & Wen, C. (2019). A hybrid Bayesian network model for predicting delays in train operations. *Computers and Industrial Engineering*, *127*(September 2017), 1214–1222. https://doi.org/10.1016/j.cie.2018.03.017

Mehryar, M., Afshin, R., & Talwalkar, A. (2019). Foundations of machine learning. In *Statistical Papers* (Vol. 60, Issue 5). MIT Press. https://doi.org/10.1007/s00362-019-01124-9

Meinshausen, N. (2006). Quantile Regression Forests. *Journal of Machine Learning Research*, *7*. https://doi.org/10.1016/j.jmva.2014.06.005

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*. Determination Press.

Nilsson, R., & Henning, K. (2018). *Predictions of train delays using machine learning*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*(85), 2825–2830. http://jmlr.org/papers/v12/pedregosa11a.html

ProRail. (2019a). *Annual Report ProRail 2019*. https://www.jaarverslagprorail.nl

ProRail. (2019b). *Impactfull disruptions of infra*. https://prestaties.prorail.nl/hinder/cDU176_Hinder.aspx

Valenti, G., Lelli, M., & Cucina, D. (2010). A comparative study of models for the incident duration prediction. *European Transport Research Review*, *2*(2), 103–111. https://doi.org/10.1007/s12544-010-0031-4

van Ammelrooy, P. (2020). *Corona of niet, ProRail voorspelt overvolle treinen en sporen | De Volkskrant*. 9-9-2020. https://www.volkskrant.nl/nieuws-achtergrond/corona-of-niet-prorail-voorspelt-overvolle-treinen-en-sporen~bfe6ee2e/

van Dale. (2019). *Van Dale Groot woordenboek van de Nederlandse taal*.

Wemelsfelder, M. (2019). *Predicting the function recovery time for railway incidents*.

Wen, C., Mou, W., Huang, P., & Li, Z. (2019). A predictive model of train delays on a railway line. *Journal of Forecasting*, *February 2019*, 470–488. https://doi.org/10.1002/for.2639

Zilko, A. A. (2017). Mixed Discrete-Continuous Railway Disruption-Length Models with Copulas. In *TU Delft University*. https://doi.org/10.4233/uuid:a551b9a2-b5da-4a51-8a3b-3d7f410d67cc

Zilko, A. A., Kurowicka, D., & Goverde, R. M. P. (2016). Modeling railway disruption lengths with Copula Bayesian Networks. *Transportation Research Part C: Emerging Technologies*, *68*, 350–368. https://doi.org/10.1016/j.trc.2016.04.018

Appendices

Appendix A

This appendix shows an overview of the features after data selection. The correlation matrix of all the numerical features.

Columns	Description	Number of categories	% Missing
IncidentID	ID of incident	790	0.0%
Locatie_bediengebied	Train route	358	0.5%
Locatie_bediengebiedsoort	Type of train route	3	0.5%
Locatie_Calamiteitenregio	ProRail region	7	0.0%
Trdlgebied	Railway traffic controller area	58	0.0%
InfraClaim	Infra logistic measure	5	0.0%
T_voorval	Time the incident occurred	790	0.0%
T_gekozeneerstevsm	Time of first diversion regulation	593	7.2%
GekozenEersteVsm_gebruiktbijteerstefasegereed	Chosen diversion regulation	267	8.4%
T_Eerstefasegereed	Time VSM fully applied	434	42.5%
T_PrognoseInitieel	Initial prognosis of incident end time	42	3.2%
T_PrognoseInitieel_Afgegeven	Time initial prognosis given	328	3.2%
T_PrognoseAangepast	Updated prognosis of incident end time	459	40.0%
T_PrognoseAangepast_Afgegeven	Time updated prognosis given	459	40.0%
T_PrognoseDefinitief	Final prognosis of incident end time	648	5.7%
T_PrognoseDefinitief_Afgegeven	Time final prognosis given	715	5.7%
aantal_MutatiesInVos	Number of mutations in traffic control operational system	7	0.0%
MutatiesInVOS_eerst	Time of the first mutation in the traffic control operational system	351	17.3%
MutatiesInVOS_eerst_afgegeven	Time of the first mutation in the traffic control operational system given	575	17.3%
MutatiesInVOS_laatst	Time of the last mutation in traffic control operational system	299	59.9%
MutatiesInVOS_laatst_afgegeven	Time of the last mutation in traffic control operational system given	313	59.9%
Logistiek_VDBs	Logistic measure to divide trains	9	6.1%
Logistiek_vsms_beoordeling	Logistic diversion regulation instruction	7	3.8%
Aantal_opheffingen	Number of cancellations	96	9.9%
Aantal_keringen	Number of diversions	37	83.7%
Aantal_uittevoerenacties_volgensvsm	Number of actions to perform according to diversion regulation	99	8.4%
Aantal_correctuitgevoerdeacties	Number of correctly performed actions	80	8.4%
SlachtofferAantalLichtGewond	Number of victims slightly injured	3	0.0%
SlachtofferAantalZwaarGewond	Number of victims badly injured	3	0.0%
SlachtofferAantalOverleden	Number of victims deceased	5	0.0%
Mate_van_Vervuiling	Degree of fragmentation	7	39.0%
AL_Terplaatsestatus	AL goes to the incident site or not	4	3.4%
AL_TerplaatseDatumTijdeta	AL estimated time of arrival incident site	633	22.0%
AL_TerplaatseDatumTijdRealisatie	AL actual arrival time at incident site	618	23.9%
FO_Werkzaamheid1	Forensic investigation activity 1	2	48.0%
FO_Terplaatsestatus	The forensic investigation team goes to the incident site or not	3	22.7%
FO_TerplaatseDatumTijdeta	The estimated time of arrival of the forensic investigation team at the incident site	320	61.6%
FO_TerplaatseDatumTijdRealisatie	Forensic investigation actual time of arrival incident site	421	48.7%
FO_DatumTijdStartWerk1	Forensic investigation start time activity 1	415	49.5%
FO_DatumTijdEindeWerk1	Forensic investigation end time activity 1	348	58.1%
FO_DatumTijdPrognoseWerk1	Forensic investigation prognosed end time activity 1	381	53.8%
ICBPloeg_Werkzaamheid1	Incident recovery team activity 1	6	37.3%
ICBPloeg_Werkzaamheid2	Incident recovery team activity 2	5	63.3%
ICBPloeg_Terplaatsestatus	Incident recovery team goes to the incident site or not	4	4.7%
ICBPloeg_TerplaatseDatumTijdeta	Incident recovery team estimated time of arrival incident site	652	19.4%
ICBPloeg_TerplaatseDatumTijdRealisatie	Incident recovery team actual time of arrival incident site	581	28.6%
ICBPloeg_DatumTijdStartWerk1	Incident recovery team start time activity 1	495	39.5%
ICBPloeg_DatumTijdEindeWerk1	Incident recovery team end time activity 1	401	51.4%
ICBPloeg_DatumTijdEindeWerk2	Incident recovery team end time activity 2	210	75.6%
ICBPloeg_DatumTijdStartWerk2	Incident recovery team start time activity 2	299	64.3%
ICBPloeg_DatumTijdPrognoseWerk1	Incident recovery team prognosed end time activity 1	467	43.0%
ICBPloeg_DatumTijdPrognoseWerk2	Incident recovery team prognosed end time activity 2	291	65.3%
MPZBegrafenisondernemer_Terplaatsestatus	Mortician goes to the incident site or not	3	14.6%
MPZBegrafenisondernemer_TerplaatseDatumTijdeta	Mortician estimated time of arrival at the incident site	594	26.8%
MPZBegrafenisondernemer_TerplaatseDatumTijdRealisatie	Mortician actual time of arrival at the incident site	462	43.7%
MPZBegrafenisondernemer_DatumTijdStartWerk1	Morticians start time of the work	461	43.8%
MPZBegrafenisondernemer_DatumTijdEindeWerk1	The mortician end time of the work	385	53.4%
MPZBegrafenisondernemer_DatumTijdPrognoseWerk1	Mortician prognosed end time of the work	426	48.2%
WDT_Terplaatsestatus	Technical team Incident Recovery goes to the incident site or not	3	49.6%
WDT_TerplaatseDatumTijdeta	Technical team Incident Recovery estimated time of arrival at the incident site	291	65.3%
WDT_TerplaatseDatumTijdRealisatie	Technical team Incident Recovery actual time of arrival at the incident site	295	64.8%
OHD_Terplaatsestatus	Government emergency services go to the incident site or not	2	27.3%
AflosReservemachinist_Terplaatsestatus	New train driver goes to the incident site or not	3	49.6%
AflosReservemachinist_TerplaatseDatumTijdeta	New train driver estimated arrival time at the incident site	321	61.5%

AflosReservemachinist_TerplaatseDatumTijdRealisatie	New train driver actual arrival time at the incident site	259	69.4%
Ovdp_Terplaatsestatus	Duty officer of the Police goes to the incident site or not	4	44.2%
Ovdp_TerplaatseDatumTijdRealisatie	Duty officer Police estimated arrival time at the incident site	306	63.4%
Ovdp_DatumTijdStartWerk1	Duty officer Police start time of activity 1	285	66.1%
Ovdp_DatumTijdEindeWerk1	Duty officer Police end time of activity 1	234	72.5%
Ovdp_DatumTijdPrognoseWerk1	Duty officer Police prognosed end time of activity 1	230	73.0%
WOP_Terplaatsestatus	Accommodation service train crew goes to the incident site or not	4	61.0%
WOP_TerplaatseDatumTijdeta	Accommodation service train crew estimate arrival time at the incident site	244	71.3%
WOP_TerplaatseDatumTijdRealisatie	Accommodation service train crew actual arrival time at the incident site	207	75.9%
Ovdb_Terplaatsestatus	Duty officer Fire brigade goes to the incident site or not	4	74.3%
Aannemer_TerplaatseDatumTijdRealisatie	The contractor goes to the incident site or not	220	74.3%
Schouwarts_Terplaatsestatus	Medical examiner goes to the incident site or not	2	95.6%
Schouwarts_TerplaatseDatumTijdRealisatie	Estimated arrival time of the medical examiner at the incident site	40	97.1%
Schouwarts_DatumTijdStartWerk1	Medical examiners start time activity 1	39	97.2%
Schouwarts_DatumTijdEindeWerk1	Medical examiner end time activity 1	29	98.5%
Schouwarts_DatumTijdPrognoseWerk1	Medical examiner prognosed end time activity 1	33	98.0%
WOR_Terplaatsestatus	Accommodation service travellers	3	80.3%
temp	The temperature in the hour of the incident	260	5.2%
temp_cat	Temperature below 0, between 0 and 20 or higher than 20	4	5.2%
mm	MM rain in the hour of the incident	2	5.2%
regen	Rain when the incident occurred yes or no	2	5.2%
sneeuw	Snow when the incident occurred yes or no	2	5.2%
onweer	Thunder when the incident occurred yes or no	2	5.2%
ijzig	Icing when the incident occurred yes or no	2	5.2%
voorval_hour	Hour the incident occurred	24	0.0%
voorval_day	Day the incident occurred	31	0.0%
voorval_dayofweek	Day of the week the incident occurred	7	0.0%
voorval_week	The week the incident occurred	52	0.0%
night	The incident occurred between 0 and 6 AM yes or no	2	0.0%
rush hour	The incident occurred between 6:30 and 9 AM or 16 and 18:30 yes or no	2	0.0%

Figure 38. Correlation matrix numerical features

Appendix B

List of features selected with Recursive Feature Elimination (Section 4.4.2) for Support Vector Machine and Neural Network.

Table 3: Wrapper RFE features for Support Vector Machine and Neural Network

Feature name
SlachtofferAantalZwaarGewond
SlachtofferAantalOverleden
FO_Werkzaamheid1
MPZBegrafenisondernemer_Terplaatsestatus
T_PrognoseAangepast
T_PrognoseAangepast_Afgegeven
T_PrognoseDefinitief
T_PrognoseDefinitief_Afgegeven
MutatiesInVOS_eerst_afgegeven
T_gekozeneerstevsm
aantal_MutatiesInVos
MutatiesInVOS_laatst
MutatiesInVOS_laatst_afgegeven
Aantal_uittevoerenacties_volgensvsm
AL_TerplaatseDatumTijdRealisatie
ICBPloeg_DatumTijdPrognoseWerk1
MPZBegrafenisondernemer_TerplaatseDatumTijdeta
MPZBegrafenisondernemer_DatumTijdStartWerk1
MPZBegrafenisondernemer_DatumTijdEindeWerk1
MPZBegrafenisondernemer_DatumTijdPrognoseWerk1
voorval_hour

Appendix C

Hyperparameter search space per model. The best parameters for every model are determined based on 10-fold cross-validation on the full data.

Lasso regression

Hyperparameters	Search space	Best parameters
alpha	[0.01, 0.02, 0.03, ..., 1.48, 1.49, 1.50]	0.1
tolerance	[1e-5, 1e-4, 1e-3, 1e-2, 1e-1]	1e-4

SVM

Hyperparameters	Search space	Best parameters
Kernel	[“linear”, “poly”, “rbf”, “sigmoid”]	“poly”
tolerance	[1e-4, 1e-3, 1e-2, 1e-1]	1e-1
Gamma	[“1 / (n_features * X.var ())”, “1/#features”]	“1 / (n_features * X.var ())”
C	[0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]	8
epsilon	[0.05, 0.1, 0.2, 0.3, 0.4, 0.5]	0.3

Decision Tree

Hyperparameters	Search space	Best parameters
Split quality measure	[“mse”, “friedman_mse”,” mae”]	“mse”
Max depth tree	[1, 2, 3, ..., 18, 19, 20]	8
Min samples for split	[1, 2, 3, ..., 18, 19, 20]	18
Max features	[1, 2, 3, ..., 18, 19, 20]	16
Min samples leaf	[1, 2, 3, 4, 5]	2

Random Forest

Hyperparameters	Search space	Best parameters
Number of estimators	[160, 180, 200, 250, 300, 400, 500]	160
Split quality measure	[“mse”, “mae”]	“mse”
Max depth of tree	[None, 8, 10, 12, 15, 20, 25, 30, 40]	12
Min samples for split	[2, 3, 4, ..., 8, 9, 10]	5
Min samples leaf	[1, 2, 3, 4, 5]	4
Max features	[“auto”, “square root of number of features”]	n

Gradient Boosted Trees

Hyperparameters	Search space	Best parameters
Number of estimators	[160, 180, 200, 250, 300]	160
Split quality measure	[“friedman_mse”, “mse”, “mae”]	“mse”

Max depth of tree	[None, 5, 8, 10, 12, 15]	12
Min samples for split	[3, 4, 5, 6, 7, 8]	5
Min samples leaf	[3, 4, 5, 6, 7, 8]	4
Max features	“auto”	“auto”

XGBoost

Hyperparameters	**Search space**	**Best parameters**
Number of estimators	[200, 300, 400, 500, 1000]	400
Learning rate	[0.01, 0.02, 0.03, 0.04, 0.05]	0.03
Max depth of tree	[None, 8, 10, 12, 15, 20, 25, 30, 40]	None
Min child weight	[2, 3, 4, 5, 6, 7, 8]	2
Gamma	[0.2, 0.3, 0.4, 0.5, 1, 1.5]	1
Colsample by tree	[0.3, 0.4, 0.5, 0.6, 0.7]	0.7

Neural Network full data

Hyperparameters	**Search space**	**Best parameters**
Layers	[1, 2, 3]	1
Neurons	[1-40, 1-40, 1-40]	29,0,0
Learning rate	[1e-1, 1e-2, 1e-3, 1e-4]	1e-3
Epochs	[5, 15, 25, ..., 1475, 1485, 1495]	615
Batch size	[10, 20, 40, 60, 80, 100]	20
Drop rate	[0.01, 0.1, 0.2]	0.2

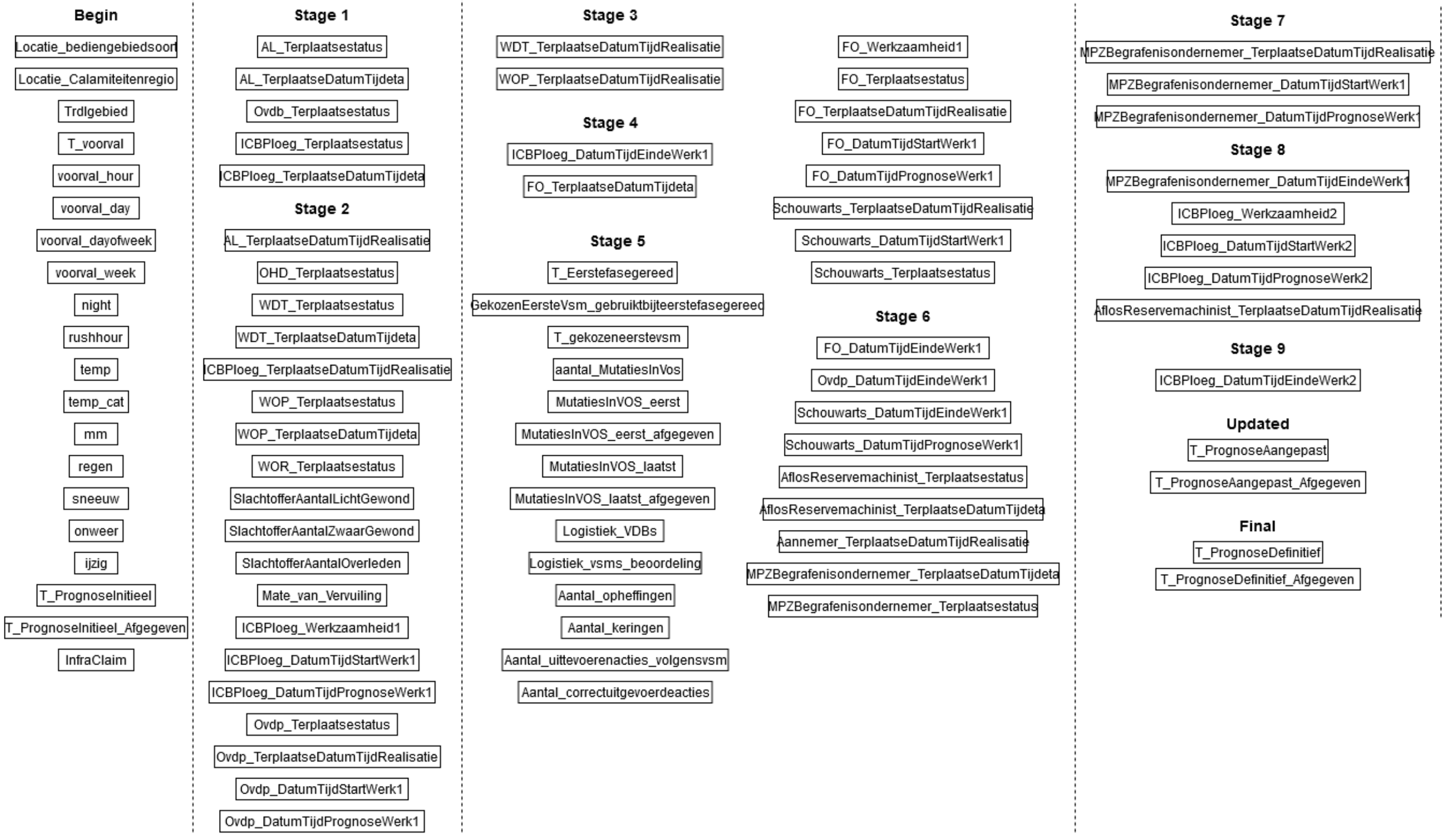
Neural Network full data (20, 29, 1)

Neural Network with feature selection

Hyperparameters	**Search space**	**Best parameters**
Layers	[1, 2, 3]	1
Neurons	[1-40, 1-40, 1-40]	29,0,0
Learning rate	[1e-1, 1e-2, 1e-3, 1e-4]	1e-3
Epochs	[5, 15, 25, ..., 1475, 1485, 1495]	615
Batch size	[10, 20, 40, 60, 80, 100]	20
Drop rate	[0.01, 0.1, 0.2]	0.2

Neural Network with feature selection (20, 29, 1)

Appendix D


Begin
Locatie_bediengebiedsoort
Locatie_Calamiteitenregio
TrdIgebied
T_voorval
voorval_hour
voorval_day
voorval_dayofweek
voorval_week
night
rushhour
temp
temp_cat
mm
regen
sneeuw
onweer
ijzig
T_PrognoseInitieel
T_PrognoseInitieel_Afgegeven
InfraClaim
Stage 1
AL_Terplaatsestatus
AL_TerplaatseDatumTijdeta
Ovdb_Terplaatsestatus
ICBPloeg_Terplaatsestatus
ICBPloeg_TerplaatseDatumTijdeta
Stage 2
AL_TerplaatseDatumTijdRealisatie
OHD_Terplaatsestatus
WDT_Terplaatsestatus
WDT_TerplaatseDatumTijdeta
ICBPloeg_TerplaatseDatumTijdRealisatie
WOP_Terplaatsestatus
WOP_TerplaatseDatumTijdeta
WOR_Terplaatsestatus
SlachtofferAantalLichtGewond
SlachtofferAantalZwaarGewond
SlachtofferAantalOverleden
Mate_van_Vervuiling
ICBPloeg_Werkzaamheid1
ICBPloeg_DatumTijdStartWerk1
ICBPloeg_DatumTijdPrognoseWerk1
Ovdp_Terplaatsestatus
Ovdp_TerplaatseDatumTijdRealisatie
Ovdp_DatumTijdStartWerk1
Ovdp_DatumTijdPrognoseWerk1
Stage 3
WDT_TerplaatseDatumTijdRealisatie
WOP_TerplaatseDatumTijdRealisatie
Stage 4
ICBPloeg_DatumTijdEindeWerk1
FO_TerplaatseDatumTijdeta
Stage 5
T_Eerstefasegereed
GekozenEersteVsm_gebruiktbijteerstefasegereed
T_gekozeneerstevsm
aantal_MutatiesInVos
MutatiesInVOS_eerst
MutatiesInVOS_eerst_afgegeven
MutatiesInVOS_laatst
MutatiesInVOS_laatst_afgegeven
Logistiek_VDBs
Logistiek_vsms_beoordeling
Aantal_opheffingen
Aantal_keringen
Aantal_uittevoerenacties_volgensvsm
Aantal_correctuitgevoerdeacties
FO_Werkzaamheid1
FO_Terplaatsestatus
FO_TerplaatseDatumTijdRealisatie
FO_DatumTijdStartWerk1
FO_DatumTijdPrognoseWerk1
Schouwarts_TerplaatseDatumTijdRealisatie
Schouwarts_DatumTijdStartWerk1
Schouwarts_Terplaatsestatus
Stage 6
FO_DatumTijdEindeWerk1
Ovdp_DatumTijdEindeWerk1
Schouwarts_DatumTijdEindeWerk1
Schouwarts_DatumTijdPrognoseWerk1
AflosReservemachinist_Terplaatsestatus
AflosReservemachinist_TerplaatseDatumTijdeta
Aannemer_TerplaatseDatumTijdRealisatie
MPZBegrafenisondernemer_TerplaatseDatumTijdeta
MPZBegrafenisondernemer_Terplaatsestatus
Stage 7
MPZBegrafenisondernemer_TerplaatseDatumTijdRealisatie
MPZBegrafenisondernemer_DatumTijdStartWerk1
MPZBegrafenisondernemer_DatumTijdPrognoseWerk1
Stage 8
MPZBegrafenisondernemer_DatumTijdEindeWerk1
ICBPloeg_Werkzaamheid2
ICBPloeg_DatumTijdStartWerk2
ICBPloeg_DatumTijdPrognoseWerk2
AflosReservemachinist_TerplaatseDatumTijdRealisatie
Stage 9
ICBPloeg_DatumTijdEindeWerk2
Updated
T_PrognoseAangepast
T_PrognoseAangepast_Afgegeven
Final
T_PrognoseDefinitief
T_PrognoseDefinitief_Afgegeven