Understanding malfunctions of smart card validators in trams by development of decision tree models

Mariëlle ten Have[1], Bas Beekmans[2], and Carolien Maas[3]

[1]*Femto Analytics, Delft, The Netherlands*
mth@femto.nl

[2]*DIKW, Nieuwegein, The Netherlands*
Bas.Beekmans@dikw.nl

[3]*GVB, Amsterdam, The Netherlands*
Carolien.Maas@gvb.nl

Abstract

Passengers can pay with an electronic smart card in every tram operational in Amsterdam. Malfunction of smart card validators leads to reduced customer satisfaction, loss of revenues and unplanned costs. Malfunction of a smart card validator is relatively rare: it fails about once every two years, which is recorded by the maintenance shop in a SAP database. The validators generate transactions and events. The events (about 200 a day per validator) are stored in the OV Chipcard database. During this project, we investigated whether it was possible to understand malfunctions by analyzing the event data generated in the 24 hours preceding those registered malfunctions. Analysis was done on over six million events generated within a four-month period (January - April 2017) by more than 1,700 validators mounted in 200 trams. The selected decision tree models showed that about 50% of registered malfunctions were related to specific events that occur in relatively high frequencies. These events signified loss of communication and/or the inability to receive GPS location information. The use of decision tree models made it possible not only to predict the malfunctions but also to get a better understanding of the root cause of the malfunctions. These insights can be used as input for improving the reliability of the smart card validators.

1. Introduction

GVB transports every day about 750,000 passengers in and around Amsterdam with about 90 metro trains, 200 busses and 200 trams. The busses and trams are all equipped with smart card validators, manufactured by Thales, to let passengers use the public transport system by checking-in or -out with their prepaid personal smart card (OV Chipcard). For trips with the metro the smart cards are needed to enter and leave the platforms.

For this paper we only analysed data from the OV Chipcard system of the 200 trams of GVB.

In the occasion of a malfunction of one or more validators in a tram, the tram driver will report the problem via mobile radio to the maintenance shop. There the malfunctions are manually registered in a SAP database. Furthermore, a work order is created and the repair is planned. The tram will be taken out of service only when none of the validators are functioning, or when a validator at one or two key positions in the tram is malfunctioning. In all other occasions the tram will continue its service for the remainder of the day.

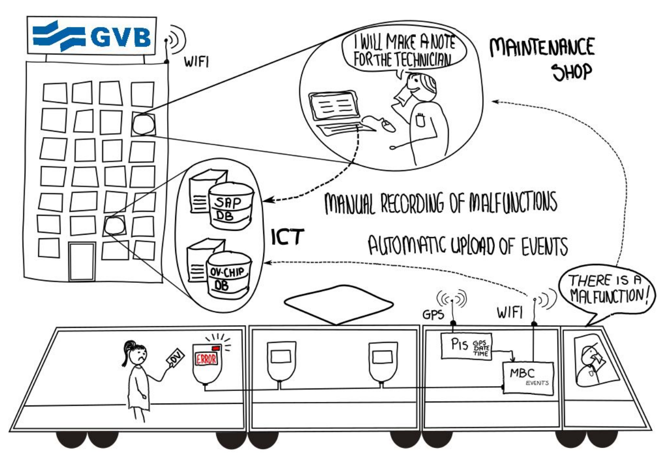

Figure 1. Information flow in case of a malfunction.

Thus, the downtime of a single smart card validator is not really an issue. Therefore the analysis was set up to understand why the validators malfunction and subsequently to prevent those malfunctions in the future. So the main goal of this analysis was to increase the reliability of the validators rather than to predict failures.

The OV Chipcard system in a tram consists of 7 to 14 validators, daisy-chained to a single Modular Board Computer (MBC). Each validator has a unique label ID which states its position in the tram and each MBC has a unique label ID that relates to the specific tram it sits in. The MBC is connected to a Passenger Information System (PIS) that has a GPS antenna. Through this connection the MBC gets information about the planned route and the exact location of the tram on its route which it puts through to all validators. The location information is essential because the validators need to know the distance traveled by a passenger between his or her check-in and check-out. Only then the travel fee can be charged. Without a (proper) GPS signal all validators will fall into a “degraded mode”. In this state a validator is unable to calculate the travel fee and the passengers can only be charged with the connection fee. Degraded mode also occurs in case of a lost connection between validator and MBC or between MBC and PIS.

The check-in / check-out events (transactions) are stored locally on the validators. Every few minutes each validator sends its stored transactions along with its system events, all enriched with its label ID to the MBC where they are collected.

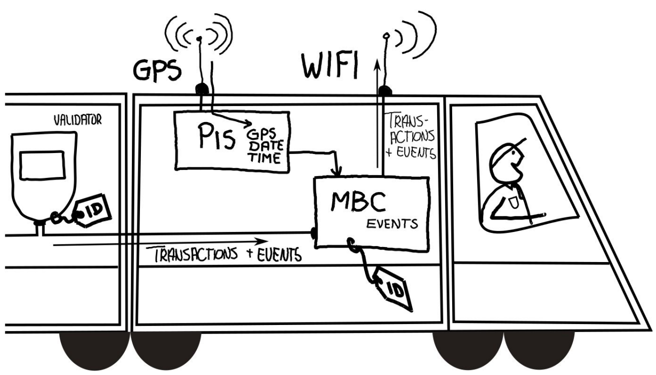

Figure 2. OV Chipcard system in a tram.

The MBC is equipped with a WiFi antenna and as soon as there is a WiFi connection with the GVB network the transactions and system events of all validators along with its own system events, all enriched with its label ID, are sent to the landside. Here the transactions are stored in the transaction database to be financially processed and the system events are stored in a separate OV Chipcard database. Furthermore the WiFi connection is used to receive software and data updates.

The main goal of this research was to investigate if it is possible to use event data of the OV-chipcard database as input for explanatory models to get a better understanding of smart-card malfunctions. Data-driven improvement of the reliability of the smart-card validators is the ultimate goal and not prediction of malfunctions, since the downtime of a single smart card validator is not really an issue.

Most reliability studies use sensor or lifetime data as input for modelling. In this study we investigate if the event data can be used to get a better understanding of smart-card malfunctions, preferably leading to actionable insights.

2. METHODS

2.1. Data Sources

For this analysis we combined the data of the SAP database, containing all the reported malfunctions, and the OV Chipcard database containing all event data. All the reported malfunctions of the validators are registered in the SAP database. In the investigated time frame (January -April 2017) there were 235 failures reported and registered in SAP. Of all these failures the date, time, vehicle ID and a global description were recorded.

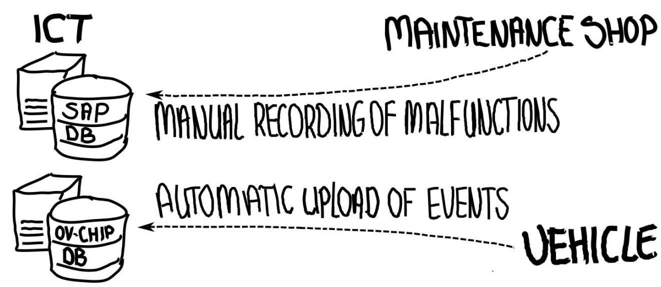

Figure 3. Data Sources.

The time recorded in the SAP database is the time of registration and not the exact time when the failure occurred. The global description is a free text field, therefore the accuracy of the description depends on the employee of the maintenance shop that created the input. As mentioned in *Meeker & Hong (2013),* the SAP database can be defined as a typical maintenance database. The data may lack important engineering information because the reporting rules and databases were designed for financial reporting rather than for answering engineering questions.

The largest database used is the OV Chipcard database. Every system event of every validator and MBC is logged here. These system events consist of information messages, errors and state changes. Because the unique label IDs are included in all events, we know exactly when (msg reportdate) which validator (equipment type 20 + Complement device ID) or MBC (equipment type 17) in which tram (Vehicle ID) has reported the event (Tagname + Tagvalue01). When a validator is functioning normally approximately 200-250 events are recorded. A sample of the database is given in Table 1.

VEHICLEID	MSGREPORTDATE	EQUIPMENTTYPE	TAGNAME	TAGVALUE01	COMPLEMENTDEVICEID
2061	3-01-17 5:18	20	ARWC01	0	11836
2027	3-01-17 5:28	20	SMLOCK	0	0
2027	3-01-17 5:28	20	VERSUS	11	0
2061	3-01-17 5:17	20	MDST04	2	10432
2027	3-01-17 5:28	20	PROMIS	0	0
2027	3-01-17 5:28	20	PRODEC	0	0
2061	3-01-17 5:17	20	EOVS04	11	10432
2061	3-01-17 5:18	20	ARWC04	0	10432
2061	3-01-17 5:17	20	MDST07	2	10662
2027	3-01-17 5:28	20	PARMIS	0	0
2061	3-01-17 5:17	20	MDST08	2	12092
2027	3-01-17 5:28	20	PARDEC	0	0
2027	3-01-17 5:28	20	PROMET	0	0
2061	3-01-17 5:17	20	EOVS08	11	12092
2027	3-01-17 5:28	20	PARMET	0	0
2061	3-01-17 5:18	20	ARWC08	0	12092
2027	3-01-17 5:28	20	METEOD	0	0
2061	3-01-17 5:17	20	MDST06	2	12998
2061	3-01-17 5:17	20	EOVS06	11	12998
2061	3-01-17 5:18	20	ARWC06	0	12998
2061	3-01-17 5:17	20	MDST03	2	13554
2061	3-01-17 5:18	20	ARWC03	0	13554
2061	3-01-17 5:18	17	GPSSTA	5	0
2061	3-01-17 5:18	17	VERSUS	11	0
2061	3-01-17 5:18	17	EODFAI	0	0

Table 1. Sample of the OV-chipcard database.

More information about three additional columns in the database was found in the technical data sheet from the manufacturer Thales *(Aubry 2004)*. In this datasheet most of the combinations of Tagname and Tagvalue were given along with the severity of this combination. In total there were 4 types of severities registered: Information, Warning, Alarm and Unknown (see Table 2). With this information we were able to enrich the data from the OV Chipcard database with the severity of the event.

Severity	Technical datasheet [3]	OV-chipcard database
Information	235	27
Warning	16	1
Alarm	104	8
Unknown	-	22

Table 2. The occurrence of the different type of Tagnames and Tagvalues found in the technical datasheet and in the OV chipcard database.

Since the data of the OV Chipcard database is generated automatically, the data quality was expected to be very high. The frequency of incomplete records was very low (less than 0,0001%, i.e. six records). Therefore we decided to ignore these records and filter them out. Furthermore, 5% of all the data was recorded twice. The exact cause was unknown so we decided to leave this data in the database. There were also new combinations of Tagnames and Tagvalues found (see also Table 2). These combinations were tagged with severity unknown.

We compared event records in the OV Chipcard database with failures recorded in the SAP database by time stamp and found correlations. We also checked if we could use only the warnings and alarms in the OV-chipcard database instead of all events. However the density of these events showed no relation with the failures found in the SAP database. Therefore, we chose to use all events.

2.2. **Data Preparation**

The timestamp recorded in the SAP database is the time of registration and not the exact time of the failure. Therefore, we chose a time frame of 24 hours prior to the SAP failure to indicate suspicious events. All events that occured within 24 hours prior to a SAP failure were marked as suspicious and got a value of S=1. All events on the same two days as the marked events but outside the 24 hour time frame were left out of the analysis. All events on all other days, not preceding a SAP failure were marked OK and received a value of S=0.

Only 235 failures were registered in SAP (S=1). These failures included 16 duplicate failures (same day and vehicle number). So for the analysis there were 219 days with suspicious events. On the other hand there were ~21.000 days (200 vehicles over a period of 115 days) with events that were marked as OK (S=0). A sample of 10% was randomly taken of the days with no SAP failure to make the ratio between SAP and no SAP failure better.

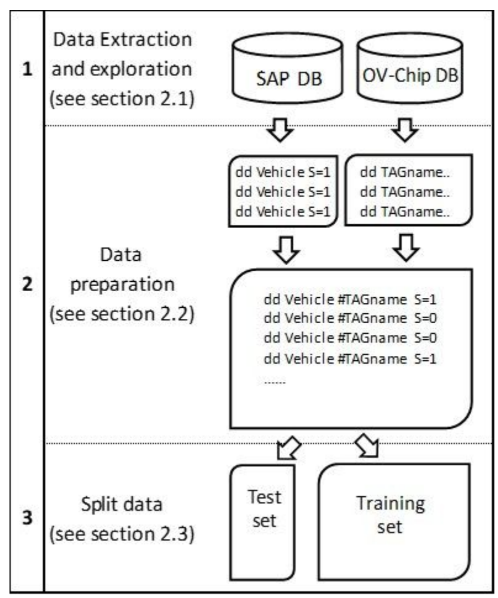

Figure 4. Block diagram of the modelling approach.

2.3. **Data Modelling**

In most literature regarding reliability analysis there is a different kind of data used to predict failures. With the use of lifetime data of one failure mode the reliability can be predicted very well (*Abernethy 2010)*. In *Meeker et al. (2013)* there are three examples described with the use of lifetime, degradation and recurrence data. The data of the OV Chipcard database however is typically event state data and most of the reported states are information or unknown.

Furthermore, since we are mainly interested in the type of events preceding a malfunction, our main research question amounts to a binary classification problem.

For binary classification, a number of machine learning algorithms can be considered, including decision trees and random forests, multinomial logistic regression, neural networks and support vector machines. We decided to use decision trees, since we aimed for a model that could be easily explained, in order to get a better understanding why malfunctions happen. Thus, prediction was not our main goal. We split the original data set into a training set for model development and a test set for studying model performance. For the training set, we randomly sampled 70% of the observations of the original data set. The remaining 30% of the observations was used in the test set.

For model development, we used the *rpart* package (V4.1-11) of Program R, V3.4.3 (R Core Team, 2017). We reduced the number of potential covariates, by only considering event types with relatively high frequencies of preceding a SAP failure. In order to avoid overfitting, we started with the default value of 0.01 for the complexity parameter (cp), and compared the results with models that had higher cp-values.

In order to develop a more robust model, we developed random forest models with the *randomForest* package (V4.6-12) in R. In order to compare the output of the random forest models with the above-mentioned decision trees, we used the same training and test sets. The random forest was based on 500 decision trees and two variables were tried at each split.

3. RESULTS

3.1 Event selection

In total there were 58 different Tagnames and -values reported in the OV-chipcard database (see also Table 2). Therefore, we focussed our analysis on the Tagnames and -values which had the highest frequency of occurrence. We calculated the number of suspicious events (S=1) per Tagname and compared this with the number of non-suspicious events (S=0). This showed that there are 6 Tagnames and -values that occurred more frequently in the S=1 event pool (see Table 3).

There was no clear description of the different Tagnames available, only a short description from the technical data sheet (*Aubry 2014*). The Tagname MTEMnn had two different values in the database, namely 0 and 1. It appeared that if MTEMnn=0 everything was function “OK” and if MTEMnn=1 that there was a “Loss of communication” (see also Table 3). Analysis of the event data also showed that the last two characters of MTEMnn were always a number, varying from 1 to 14. Since there were at most 14 validators in 1 tram, we supposed that this number was the validator number. Normally MTEMnn=0 appeared 19.6 times per 24 hours in the database, however when there was a failure reported this value increased to an average of 29.5 times per 24 hours (Table 3).

Tagname	Tagvalue	Severity	Description	Average number of events S=1	Average number of events S=0
MTEMnn	0	Information	OK	29,5	19,6
MTEMnn	1	Information	Loss of communication	19,6	6,9
GPSSTA	1	Unkown	GPS is "ON"	37,3	8,1
GPSSTA	5	Unkown	GPS is "Out of order"	39,9	14,0
PARDEC	1	Alarm	Decoding error	8,0	0
VERSUS	0	Information	Version number of PVU	19,0	0

Table 3. The Tagnames correlated with SAP failures.

The other Tagname which appeared frequently in the event database was GPSSTA. This Tagname could have the value 1 or 5. The exact severity of these Tagname was unknown since it did not appear in the technical datasheet (*Aubry 2014*). The explanation GPSSTA=1 and GPSSTA=5 was retrieved from the latest release notes. If GPSSTA had value 1 than the GPS is “ON” and if GPSSTA had value 5 the GPS is “Out of order”. When the smart card validators were functioning well then the GPSSTA=1 event was normally reported 8.1 times per 24 hours. When there was a malfunction reported this average frequency of occurrence increased to 37.3 times per 24 hours.

The Tagname with PARDEC = 1 had severity “Alarm” with description “Decoding Error”. This Tagname was not very frequently in the event database and only appeared when there was a real SAP failure. Only 4 of 219 malfunctions recorded in the database could be explained by this Tagname. The PARDEC=1 was the only alarm found in the database which showed a relation with malfunctions of the validators of the in total 104 alarms which were described in the technical datasheet of Thales *(Aubry 2004).*

The last Tagname which also showed a clear correlation with the SAP failures was VERSUS. Description of VERSUS in the technical datasheet was “Version number of PVU” and had severity “Information”. Normally this Tagname had value 11 (which is the current version installed on the MBC). However, when the VERSUS had value = 0 then a malfunction of the validators was reported. Inspection of this event showed that only 5 out of 219 malfunctions could be explained by this Tagvalue. Four of these malfunctions were the same as with PARDEC=1.

The frequencies of occurrence of the six Tagnames, described in this section, were considered as input variables for the developed models.

3.2 Decision tree model

The training set contained 1,632 samples with 153 failures (= 9.38%).

Most splits in the decision tree are related to MTEMnn. The first split in the decision tree is if more than 182 times MTEMnn=1 ("Loss of communication") is registered then 94.1% of the population has a failure (S=1). In total there are 17 records (n=17) which fulfill this condition and 16 of these records have a registered failure (S=1).

The second split is MTEMnn=1 < 182 and MTEMnn=1 >= 6.5. This means that if the validator has lost communication with the MBC between 6.5 and 182 times a day then there is a 27.4% chance that a SAP failure occurs as well.

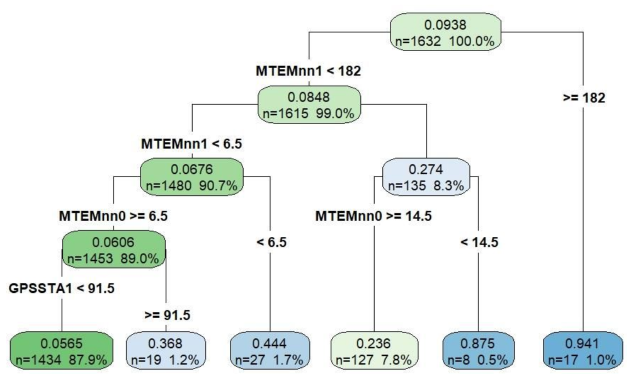

Figure 5. Decision tree model.

The second important variable was GPSSTA. This variable also switches frequently between GPSSTA=1 ("GPS is ON") and GPSSTA=5 ("GPS is out of order"). The last split of the decision tree shows that if GPSSTA=1 is recorded less than 91 times a day then there is only a 6% percent chance of a SAP failure.

Two variables (VERSUS and PARDEC) did not appear in the decision tree. The Tagname VERSUS gives information about the version number of the installed PVU software. Normally the version had value 11 in the original dataset. However, some smart card validators in 3 trams showed for a short period of time that software version 0 was installed. The deviant software version occured in combination with only a very small number of SAP failures (three in the training set), which explains the absence of VERSUS in the decision tree model.

With the use of the validation set (30% from the original data set) the decision tree model could be validated. The results are plotted in a ROC-curve (Figure 6) and a confusion matrix (Table 4). The ROC-curve shows the relationship between the false positive rate and the true positive rate. As such, the ROC-curve can be used to compare performances of various types of models The area under the curve of the ROC chart is 0.71 which is acceptable.

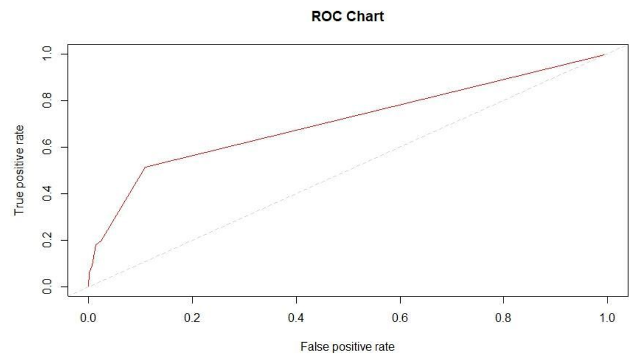

Figure 6. The ROC curve decision tree model.

The confusion matrix showed that 34 of all 66 SAP failures that combined with S=1 events were predicted right. In 69 times the model predicted a SAP failure while no SAP failure was recorded (False Positives). There were 32 registered SAP failures which were not predicted by the decision tree model (False Negatives).

	SAP no failure (S=0)	SAP failure (S=1)
Predicted No failure	565 (True negatives)	32 (False Negatives)
Predicted Failure	69 (False positives)	34 (True Positives)

Table 4. The confusion Matrix.

3.3 Random forest

The output of random forest models is harder to interpret than the output of decision trees. However, the variable importance plot can be used to determine which input variables are most important in the selected random forest. Figure 7 shows the variable importance plot for the selected random forest model, with Mean Decrease Gini on the horizontal axis. The Mean Decrease Gini is a measure of how each variable contributes to the resulting homogeneity of the nodes and leaves in the selected random forest. . Thus, the variable with the highest Mean Decrease Gini is the most important, since it contributes the most to a better classification of SAP failures occurring or not. For the random forest this is variable MTEMnn=0.

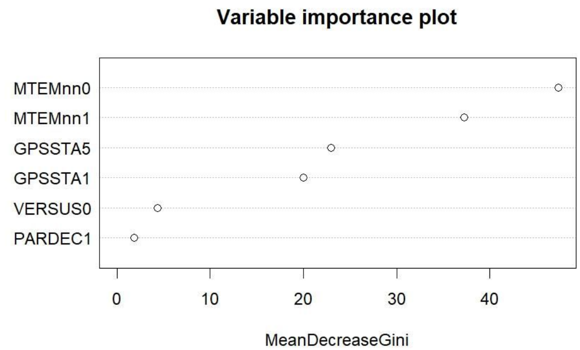

Figure 7. Variable Importance Plot of the random forest.

The results of the random forest are validated with the use of a validation set. The results of this validation are plotted in a ROC-curve. The area under the curve is 0.74 which shows that the random forest model performs somewhat better than the decision tree model.

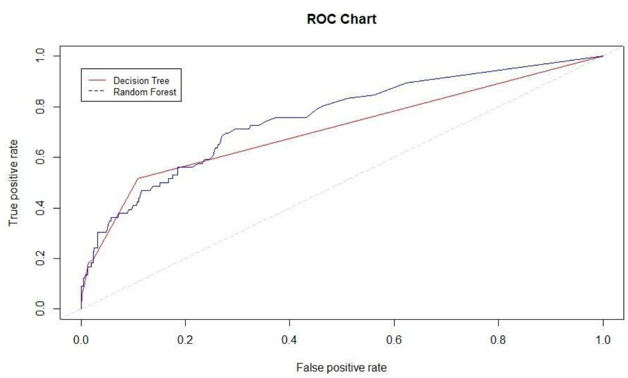

Figure 8. ROC Chart of random forest model.

4. DISCUSSIONS & CONCLUSIONS

The performance of a random forest is normally better than a decision tree. However the interpretation and visualisation of a random forest is more difficult because the output of a random forest is the majority vote of all trees. The output of the random forest showed that MTEMnn=0 is most important variable however it is not clear at which level this variable can give a failure. For instance when a tram is functioning normally, MTEMnn=0 can occur 20 times a day (see Table 3). In contrast, the decision tree generate very easy to understand rules and as stated in (*Saitta 2005*) the tree brings out meaning to data.

Our main goal was to understand why the SAP failures occurred and the simple rules generated by the decision tree can help to improve the reliability of validators.Therefore, the ratio between the true positives and false negatives is very important. This gives an indication which percentage of all malfunctions can be explained by the model. In this model this is 55%. So the reliability could be improved with 55%.

The model is not used for prediction and therefore the high number of false positives is not really an issue. However, by using the decision tree rules of the selected model, we could study the number of 69 false positives in the validation set in more detail (Table 4). It turned out that most of the false positives were validators which did have a SAP failure two or three days after the suspicious events. Thus, either the time window of 24 hours prior to the SAP failures was too short or not all failures of the validators were registered in SAP. Furthermore, the test set showed that the predicted failures were all of one type of trams (Siemens Combino). This might be due to the RIS/MBC of this type of trams not being switched off directly when the tram is taken out of service.

The strength of decision trees is that they generate easily understandable rules. The decision tree used for the smart card failures provided insights to GVB which can be used as input for improving the reliability.

REFERENCES

Abernethy, R.B. (2010) *The New Weibull Handbook.*

Aubry, P. (2004) *SDOA_DIS_Supervision_B1 (events en alarms).xls.* Technical Datasheet Thales.

Meeker, W.Q., & Y. Hong (2013). *Reliability Meets Big Data: Opportunities and Challenges*. Iowa State University Statistics Preprints. 82 http://lib.dr.iastate.edu/stat_las_preprints/82

R Core Team (2017). *R: A language and environment for statistical computing.* R foundation for Statistical Computing, Vienna, Austria. URL: https://www.R-project.org.

Saitta, S., Raphael, B. and Smith, I.F.C (2005). *Data mining techniques for improving the reliability of system identification.* Advanced Engineering Informatics, Vol 19, No 4, 2005, pp 289-298. doi:10.1016/j.aei.2005.07.005.

Thijn, van E.M., (2009). *Functionele beschrijving GVB OV-Chipkaart system.* GVB-rapport, Amsterdam, The Netherlands.

BIOGRAPHIES

Mariëlle ten Have has a Master degree in Mechanical Engineering and Transport, Infrastructure & Logistics from the Technical University Delft. For more than 10 years she was working in the railway industry as information analyst. Nowadays she is partner and senior data analyst of Femto Analytics Delft.

Bas Beekmans has a PhD degree in Biology, obtained from Groningen University, the Netherlands. He has been analysing and modelling data for more than 10 years. Since 2016, Bas works as data scientist at DIKW, Nieuwegein. In addition, he provides training in data science at DIKW.

Carolien Maas has a BSc degree in biochemistry from the Hogeschool van Amsterdam, the Netherlands. For two years she was a Research Analyst followed by a 16 year career in ICT. She is currently a Reliability Engineer at the rail division of GVB Amsterdam, Netherlands where she analyzes failure risks of tram systems and leads investigations into the root cause of these failures.