

The Application of Bayesian Networks in the Domain of Theft Alarm Analysis

Alex Bijsterveld
S0520195
Artificial Intelligence
Radboud University Nijmegen

MSc Thesis in Artificial Intelligence

Supervisors Radboud University:

dr. ir. Johan Kwisthout
Donders Institute for Brain, Cognition, and Behaviour
Radboud University Nijmegen

dr. Iris van Rooij
Department of Artificial Intelligence
Donders Institute for Brain, Cognition, and Behaviour
Radboud University Nijmegen

Supervisors Allsetra B.V.:

Steven Hoen
Simon van der Linde
Raymond van Dorresteyn

External examiner:

dr. Marina Velikova
Department of Model-Based System Development
Institute for Computing and Information Sciences
Radboud University Nijmegen

March 2013

Abstract

In connection with the application of Bayesian networks in several domains like medical diagnoses and weather forecasting, this thesis introduces the use of Bayesian networks into the field of theft alarm analysis. It is investigated if Bayesian networks could be able to assist in making judgments about incoming alarms from vehicles using historical alarm data available from those same vehicles. The model is tested in the real world environment of Allsetra, which provides track-and-trace solutions for vehicles using built-in electronics. The test results give an insight on what percentages of false alarms could be filtered out using Bayesian networks.

Acknowledgements

First, I want to thank Johan Kwisthout and Iris van Rooij for guiding me during my research and the writing of my thesis. Second, I want to thank Allsetra for the opportunity to do my master's internship at their place. Within Allsetra, my special thanks go out to Simon van der Linde en Raymond van Dorresteijn for their technical support and their helpful assistance. I also want to thank Steven Hoen for his support and for bringing me and Allsetra together.

Contents

1	Introduction	5
1.1	Background	5
1.2	Medical domain	5
1.3	Industrial domain	6
1.4	Research setting	7
1.5	Scientific aims and relevance	7
1.6	Overview	9
2	Preliminaries	10
2.1	Bayesian networks	10
3	Modeling	13
3.1	Motivation	13
3.2	Network types	13
3.3	Variables	14
3.3	Static Bayesian network specifications	15
3.4	Questionnaire for probabilities	17
3.5	Prior Information	21
3.6	Dynamic Bayesian network specifications	21
3.7	Software	21
3.8	Data Allsetra	22
4	Validation	23
4.1	Static Bayesian network results	23
4.2	Dynamic Bayesian network results	25
5	Discussion	27
5.1	Allsetra	27
5.2	Innovations	28
5.3	Future	29
5.4	Conclusion	29
A	Questionnaire	32
A.1	Original questionnaire for domain experts in Dutch	32
A.2	Original questionnaire for domain experts in English	38
B	Source code	44
B.1	Static network	44
B.2	Dynamic network	58

1 Introduction

Bayesian networks are formal models for representing, and reasoning with, uncertain information. Such models have proven useful as decision support systems in several medical situations where information at hand was uncertain and decisions nevertheless needed to be made. They have also been employed in other areas where diagnoses are to be made by domain experts. However, they have never been applied in the domain of theft alarm analysis. This domain differs from, e.g., the medical domain, as the genuine alarms (in the medical domain the persons with the disease) do not have specific symptoms to distinguish them. In this thesis, the Bayesian network formalism is used to build a system that can assist in diagnosis concerning the validity of theft alarms of vehicles. To achieve this, certain challenges need to be dealt with. At first, patterns of reoccurring circumstances need to be found in the historical data of earlier alarms to make decisions about a new incoming alarm. Second, expert knowledge is needed to make estimations on the probability that an alarm is valid or false. At last, valid alarms should never be diagnosed as false, as this could result in a stolen vehicle.

1.1 Background

Many companies use call centers to help and assist customers with problems. Large companies with a lot of customers will need a lot of employees to cover for these large numbers of customers. To decrease the number of customers calling to the support center, and thus the number of employees needed, some companies create an online support system on their website. Questions that are often asked by customers are gathered and explained on the website, resulting in fewer phone calls to the call center; sometimes, techniques from artificial intelligence are used to parse natural language queries.

Not all call centers, however, are set up for customers with questions that need to be answered. Some call centers receive automatic messages (e.g., through SMS) with warnings about occurring situations. An example of such message system can be found at some water boards in the Netherlands. SMS-alarms are sent out when the flow rate of the water becomes too high. These specific type of call centers only become active when an alarm message is received. Such an alarm can be sent from all sorts of devices and with all sorts of reasons. At some of these call centers, a lot of these alarms could be triggered, with no actual emergency situation occurring. Then, only a small number of alarms would be truly genuine. Because of the large number of false alarms, employees could become less focused and the risk of making mistakes in case of a genuine alarm could increase. To decrease the number of false alarms arriving at the call center, artificial intelligence techniques could be used to filter these alarms beforehand. With these techniques human methods to carry out certain tasks could be mimicked or even improved. This way, part of the work that needed to be done by employees could now be done by a computer.

In this thesis, a particular technique in artificial intelligence – namely, Bayesian networks – is used to aid in diagnosis of the validity of an incoming alarm. Ben-Gal (2007) shows that a Bayesian network, also known as a belief or probabilistic network, could be a suitable model to estimate the probability of the validity of an alarm. Especially in an uncertain domain where it is not self-evident whether an incoming alarm is true or false.

1.2 Medical domain

Earlier research shows the use of probabilistic networks for diagnosing certain medical situations. A medical situation can, at first sight, in some way, be compared with diagnosing a theft alarm, because experts in both domains intent to recognize certain indications that usually denote the presence of a disease or a theft. Later on in this chapter I will also show an

important difference between the two domains, which makes this novel research of particular applied interest. In the next paragraphs three examples of research concerning the use of probabilistic networks in the medical domain will be reviewed.

At first, De Bruijn, Schurink, and Hoepelman (2000) introduced a probabilistic and decision-theoretic system that aims to assist clinicians in diagnosing and treating patients with pneumonia in the intensive-care unit. Its underlying probabilistic network model includes temporal knowledge to diagnose pneumonia on the basis of the likelihood of micro-organisms causing disease combined with the symptoms and signs actually present in the patient. An optimal antimicrobial therapy is selected by balancing the expected efficacy of the treatment against the spectrum of antimicrobial treatment. Expert knowledge was used as a basis for the models.

Second, Van der Gaag, Renooij, Witteman, Aleman, and Taal (2002) came up with a decision-support system for patient-specific therapy selection for oesophageal carcinoma. The system consists of a probabilistic network that describes the characteristics of oesophageal carcinoma and the pathophysiological processes of invasion and metastasis. The probabilities required for the network were retrieved from experts using a new method that combines the ideas of transcribing probabilities as fragments of text and of using a scale with both numerical and verbal anchors for marking assessments. Using data from 185 patients to test the quality of the probabilities obtained, they found that, for 85% of the patients, the network yielded the correct outcome.

At last, Wasyluk, Onisko, and Druzdzal (2001) described a probabilistic causal model for diagnosis of liver disorders named HEPAR II. It is based on expert knowledge combined with clinical data retrieved from medical records. The Bayesian network captured the causal interactions among various risk factors, diseases, symptoms, and test results. Its main applications were to assist in diagnosing and to train beginning diagnosticians.

In all three examples of applications of Bayesian networks in the medical domain three interesting points come up. First, expert knowledge is gathered to describe dependencies and estimate probabilities used to model chances of occurrence of a certain disease. Second, all three Bayesian models are used to assist in a diagnosing situation. Third, they all assist in diagnosing the *presence* of a medical condition, rather than the *absence* of such a condition.

1.3 Industrial domain

Besides the application of Bayesian networks in the medical domain there also have been some applications of Bayesian networks providing diagnoses in industrial circumstances, which are also connected with the research in this thesis. A lot of AI research is done in academia using standard problems and datasets. It is a challenge to apply an AI technique, like Bayesian networks, in an industrial setting. Little guideline support for modeling systems in an industrial context and little theory of using Bayesian networks in an industrial setting is available. In the next paragraphs four examples of research concerning the use of probabilistic networks in an industrial setting will be reviewed.

At first, Dey and Stori (2005) created a Bayesian belief network to diagnose the root cause of process variations in a production machining environment. They used multiple process metrics from multiple sensor sources in sequential machining operations to identify this root cause and provided a probabilistic confidence level of the diagnosis.

Second, Hommersom and Lucas (2010) argue that conventional control engineering solutions increasingly fall short. Conventional control engineering techniques assume that a physical system's dynamic behavior can be completely described by means of a set of equations. On the one hand, this shortcoming exists due to the modern systems that are often of a high complexity and incompletely understood; on the other hand, it exists due to the observations obtained from sensors during runtime that give an incomplete picture. They state

that probabilistic reasoning would allow one to deal with these sources of incompleteness, yet in the area of control engineering such AI solutions are rare. In their paper they show that it is possible to use a Bayesian network to control a complex system's behavior.

Third, Cofiño, Cano, Sordo, and Gutiérrez (2002) chose to use a Bayesian network to model spatial and temporal dependencies among a network of meteorological stations. The main reason to support this decision was that standard approaches like analogue techniques and neural networks do not consider all available information. Cofiño et al. illustrate the efficiency of the use of Bayesian networks by obtaining precipitation forecasts for 100 meteorological stations.

At last, Kennett, Korb, and Nicholson (2001) examined the use of Bayesian networks for predicting sea breezes. They developed some networks based on expert elicitation and some learned by two machine learning programs. The results are compared with a pre-existing rule-based system. The Bayesian networks clearly outperformed the rule-based system.

These previous studies show that Bayesian networks are often used for diagnostic purposes and they perform well in a wide range of industrial environments. In this thesis, the application of Bayesian networks is extended to a novel industrial domain: diagnosing the validity of incoming theft alarms.

1.4 Research setting

The research on which this thesis is built was mainly executed in the environment of Allsetra. Allsetra (<http://www.allsetra.nl/>) is a company that provides complex track and trace solutions for vehicles, boats and operating equipment in construction using built-in electronics. These built-in electronics are packed in a box which is hidden in the vehicle. This box registers all sorts of information like GPS coordinates, driving speed, if the engine of the vehicle is on and if the vehicle is moved. An important functionality of this box is the ability to send an alert to the Service Center when something is happening to the secured object that should not happen. The object could for example be moved at a time it should not move or it could be moved to a place where it should not go. When the Service Center receives such an alert it needs to find out whether the object is being stolen and the police needs to be informed or that the alert is false and no further actions need to be taken.

Many incoming alarms are found to be unnecessary or redundant. Based on human interpretation and intervention, some of the alarms can be eliminated in advance. Reoccurring events or a certain pattern of alarms can, for example, be interpreted as false alarms. The remaining alarms need to be handled by the Service Center which gives cause to an extensive load of work and a lot of customers are unnecessarily contacted.

Allsetra is interested in an intelligent system that is able to incorporate human interpretation and intervention into the model to filter out some of the easy cases so that the focus of the Service Center is with the harder cases that need to be checked upon. As a result, the amount of false emergency calls to the customers and the amount of messages needed to be interpreted by the co-workers of the Service Center should be considerably reduced. A system like this will have to deal with uncertain information and combined with other information it should give an outcome indicating the probability that an alarm signals a genuine alarm situation.

1.5 Scientific aims and relevance

The new domain this research is focusing on could be compared to the medical domain reviewed earlier in this introduction. A genuine alarm could be compared with a disease that a patient could be diagnosed for. By taking several symptoms and signs into account the probability that a specific disease is at hand is calculated. However, a genuine alarm does not

have specific symptoms and signs and it does not occur often enough to find reliable regularities. Therefore, genuine alarms cannot easily be recognized. The number of alarms that could be genuine however, could be decreased by recognizing the false alarms. False alarms occur very often and are caused by human beings that more than once do not follow the agreed rules. Another difference between the medical and the current domain is that a Bayesian model in the medical domain is intended to fit on a large group of people. The idea is that, when every person in that group would be infected with a certain disease, every person's symptoms would be comparable. However, in the current domain this idea does not hold. The circumstances (in the medical domain the symptoms) in which a false alarm (the disease) occurs are different for each vehicle (the people). When transposed to the medical domain, this could mean that for a certain disease one person has the symptoms high body temperature and high blood pressure, while another person has the symptoms low body temperature and low blood pressure.

The goal of this research project is to investigate if a Bayesian network could be used to aid in diagnosis of incoming theft alarms. Practically, in the Allsetra environment, this means that the number of messages being handled by the co-workers of the Service Center can be reduced without missing any genuine alarm situations. To achieve this, some important research questions will need to be answered.

The first question is about the unique necessity in this domain to deal with the different patterns of circumstances in which alarms occur for each vehicle. An example of such a pattern could be, if two alarms from the same vehicle occurred at the same time and on the same day of the week. The question will be if it is possible to create a Bayesian model that fits every vehicle, while for each vehicle different instantiations of patterns apply.

The second research question is about the probabilities that will be used by the Bayesian network to make judgments on incoming alarms. Because there is no objective information available about these probabilities, experts will have to give an insight in the domain of judging alarms. Keeping in mind that circumstances of alarms differ for each vehicle, the question will be if the experts still can provide enough information that can be used by the Bayesian network to make a correct probability judgment on the falseness of an incoming alarm.

The last research question is about the constraint of Allsetra to avoid filtering out any true alarm case, which is very important in this domain. Such a constraint, where a certain classification cannot be missed, could also exist in other domains, but for some the consequences of breaking it could be worse than for others. For example, imagine the difference between a wrong weather forecast and a life threatening disease that is not recognized. Therefore, in this domain, an important research question is: is it feasible to have a Pareto optimal working network? Pareto efficiency or Pareto optimality was introduced by Pareto (1896) in economics, but it also has its applications in engineering, e.g., where there are several design objectives, some of which may be competing, and the goal is to choose a solution that maximizes benefits subject to the existing constraints. In the context of this research project, Pareto optimality means that as many false alarms as possible will be filtered out, without losing any true alarm cases. Because there will always be a finite possibility of having a valid alarm being judged as a false alarm, this definition of Pareto optimality should be slightly adjusted to: as many false alarms as possible will be filtered out, with a negligible chance of losing any true alarm cases. To be one hundred percent certain that no valid alarms would be filtered out, all alarms should be diagnosed as valid. As this will not lead to a reduction of alarms being handled by the Service Center, it is not a satisfying solution for Allsetra.

To answer these research questions several challenges will need to be tackled. To test a network it will need to be implemented in the already existing environment of Allsetra.

Several domain experts will be questioned to gather knowledge about the probabilities used and the importance of information gathered. The test results of the different networks that were created during this project will be compared. What is their performance on the true positive alarm cases? How well do they filter out the alarm cases that actually are not an alarm (the true negatives)? How many cases are judged as an alarm, but actually are not (the false positives)? And how many false negative judgments are made?

1.6 Overview

The remainder of this thesis is structured as follows. In Chapter 2, the Preliminaries section, the basics of Bayesian network theory are explained for readers with little knowledge about this subject. In Chapter 3, the Modeling section, the choices of design are made clear and the way parameters and data are retrieved is shown. In Chapter 4, the Validation section, results of the research are presented and lastly, in Chapter 5, the Discussion section, the results are discussed and the research questions are answered.

2 Preliminaries

In this section, a concise introduction is provided for readers with little knowledge about Bayesian networks. For a more thorough discussion of this concept the reader is referred to standard textbooks like Pearl (1988) or overview articles such as Pearl and Russell (2000).

2.1 Bayesian networks

Bayesian networks, belonging to the family of probabilistic graphical models, are used to represent knowledge about an uncertain domain and were first developed by Pearl. A Bayesian network consists of a graphical structure that models a set of stochastic variables, the conditional independencies among these variables, and a joint probability distribution over these variables. The graphical structure consists of nodes and edges, where the nodes represent random variables and the edges between the nodes represent probabilistic dependencies among the corresponding variables. Statistical and computational methods are used to estimate these conditional dependencies. Figure 1, adapted from Pearl (1988), is an example of such a Bayesian network.

In this example the grass can be wet due to the sprinkler or the rain. The probability that the grass is wet, given that the sprinkler is activated and it rained, is $P(W=\text{true} \mid S=\text{true}, R=\text{true}) = 0.99$. When the sprinkler is off and it did not rain, the probability of wet grass is $P(W=\text{true} \mid S=\text{false}, R=\text{false}) = 0$.

Different types of Bayesian networks exist. Diard, Bessière, and Mazer (2003) proposed a general-to-specific ordering of probabilistic modeling formalisms. The more general purpose models are Bayesian Networks, Dynamic Bayesian Networks, Recursive Bayesian Estimation, Hidden Markov Models, Kalman Filters, and Particle Filters, whereas the more problem oriented models focused on the field of robotics consist of Markov Localization, Decision Theoretic Planning, Bayesian Robot Programming, and Bayesian Maps.

Bayesian networks are a primary method for dealing with probabilistic and uncertain information. They combine the theory of (independency in) probability distributions and the theory of graphs in order to yield efficient representation of probabilistic (in-)dependencies in stochastic variables. Dynamic Bayesian networks are an extension of Bayesian networks that can also deal with stochastic processes that change over time. Recursive Bayesian Estimation is the generic denomination for a class of numerous different probabilistic models of time series. Examples of this class are ‘filtering’, ‘prediction’, and ‘smoothing’. Hidden Markov Models and Kalman Filters are specializations of this Bayesian Filtering where ‘filtering’ refers to determining the distribution of a latent variable at a specific time, given all observations up to that time. Particle Filters may be seen as a specific implementation of this Bayesian Filtering where a set of differently weighted samples of the distribution (particles) is used to allow for approximate ‘filtering’.

Markov Localization is Bayesian Filtering extended with control variables used in robotics where observation and movement play a large role. Decision Theoretic Planning is used in robotics to model a robot that has to plan and to execute a sequence of actions. Bayesian Robot Programming is applied to mobile robotics and Bayesian Maps are a generalization of Markov Localization.

Some of these methods are particularly suited for specialized tasks, like filtering, which does not apply here; hence we need to use a more general network structure. When designing this structure, four main aspects could be considered. The first important decision to make is what variables to use in the network. These variables need to give relevant information concerning the truthfulness of the alarm. A sensitivity analysis, described in detail by Saltelli (2004), can help to find out if the chosen variables are the most important

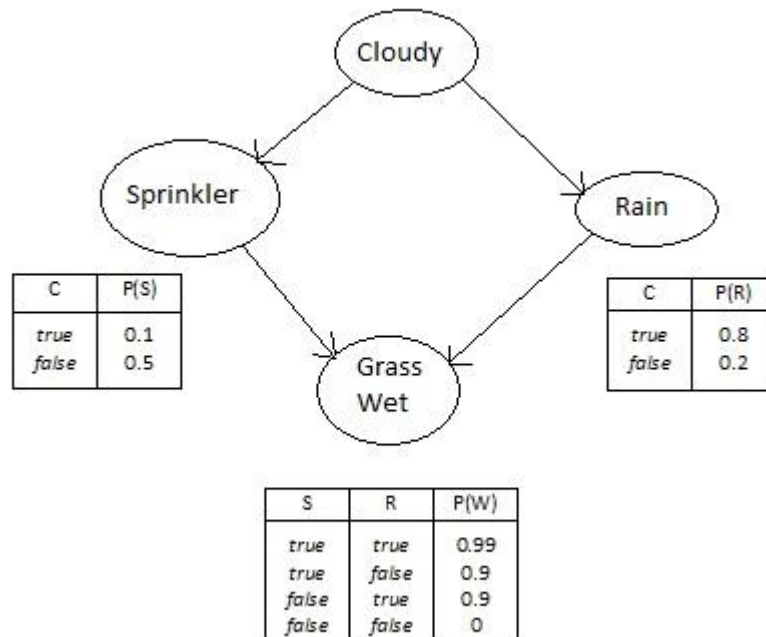


Figure 1: Example of a Bayesian Network with its joint probability distribution. Adapted from Pearl (1988).

ones. With this analysis it is possible to measure how much influence each variable has on the probabilities of other variables. In case of a Bayesian decision support system there probably are few important variables that influence the outcome of the actual classification variables (Druzdzel, 1994). Hence, with the analysis one mainly considers the influence of those few variables on the classification variables.

The second aspect to consider is if the variables in the network need to be discrete or continuous. The choice between discrete or continuous variables depends on the type of information that is represented by each variable. A continuous variable can take on any value between two specified values, while a discrete variable can only take on a range of predefined values. As a consequence, the probability distributions will differ. The continuous probability distribution is described by an equation or formula, while the discrete probability distribution can be described in a tabular form. Continuous variables will need more computational power, but can be converted to discrete variables by making intervals.

The third main aspect are the (in)dependence relationships between the variables. Variables which are not connected are conditionally independent, while edges between two nodes represent conditional dependencies.

The fourth aspect to consider is the determination of the Conditional Probability Tables (CPTs) of the variables in the network. These CPTs represent the joint probability distributions of the variables in the network.

Another important aspect is the consideration to make use of changes over time or not. When variables are only represented in the model at one moment in time it is called static. When multiple copies of the variables in the model are represented at different points in time, the model is called dynamic, as it captures not only static information, but also how the posterior probability distributions change over time.

The computational complexity of the model should also be taken into account. Continuous variables could lead to longer computation times during the inference on the network, due to the marginalization that needs to be done over real-valued domains. When the full expressiveness of Bayesian networks is used it will result in a high complexity. More arcs and more variables lead to more parameters that need to be calculated. To reduce computation times some constraints could be imposed, for example with regards to independence relationships between variables. This could reduce the complexity.

3 Modeling

3.1 Motivation

At this point, one could suggest using logistic regression instead of a Bayesian network. With a sample of the data, including the values for the output variable, a logistic regression model could be learned, predicting the validity of a new incoming alarm. However, there are four main reasons why Bayesian networks still are considered more suitable in this setting.

At first, it could perhaps not be assured, for the set of variables used in this research, that the information is always available. In that case, prior information needs to be used and then a Bayesian network would be a suitable model. The second reason is the intention to make use of temporal information of the alarms. A dynamic Bayesian network, in that case, is more useful than a logistic regression model. The third reason to not choose logistic regression could best be explained with an example. When a lot of false alarms occurred on Monday, a logistic regression model would give a new alarm on Monday a large probability of being false. However, in this domain, experts indicated that more information should be used to come up with a probability, because, this way, a genuine alarm on Monday could too easily be considered false. The experts would rather see that, besides the day the alarm occurred, also another piece of information would be used to diagnose if an alarm is false. The last reason to not use logistic regression is that the data available is not labeled, which prevents learning the parameters of a logistic regression model. Bayesian networks, on the other hand, can be built manually, based on expert knowledge.

3.2 Network types

The first decision to make is which type of Bayesian network to use. A specific type of network is difficult to pick as the domain the network is applied to, is yet unexplored. Specific types of networks have constraints for which, at the moment, it is unclear if they can be met. An example of this would be a Naïve Bayesian Classifier, described by Lewis (1998). It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. In the Allsetra case, it would be assumed that the variables that provide evidence to decide whether an alarm is false or not, would be independent of each other, given knowledge whether or not the alarm is false. This is an assumption that cannot be made, because at the moment there are no reasons to believe that the chosen variables are indeed independent of each other and thus the assumption may not be valid. Therefore, the more general type of network is chosen for. Follow-up research could evaluate more specific types of networks using the outcomes of this thesis.

In this study, we compared two different Bayesian networks. The first one is a static network which uses the evidence of a new alarm of a box to determine the validity of that alarm (Figure 2). The second network is a dynamic network which has the static network as a base and also uses the outcome of earlier alarms to determine the validity of a new alarm (Figure 3). A dynamic network could provide a more realistic computation, because it takes the change over time into account. However, this type of network could be more complex than the static network, which could result in longer computation times. Later, the network models will be discussed in more detail, but first the variables of the networks are described.

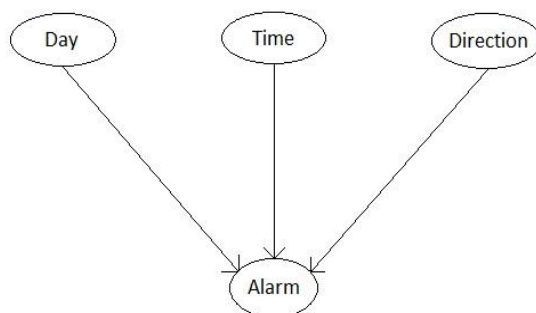


Figure 2: Static Bayesian network

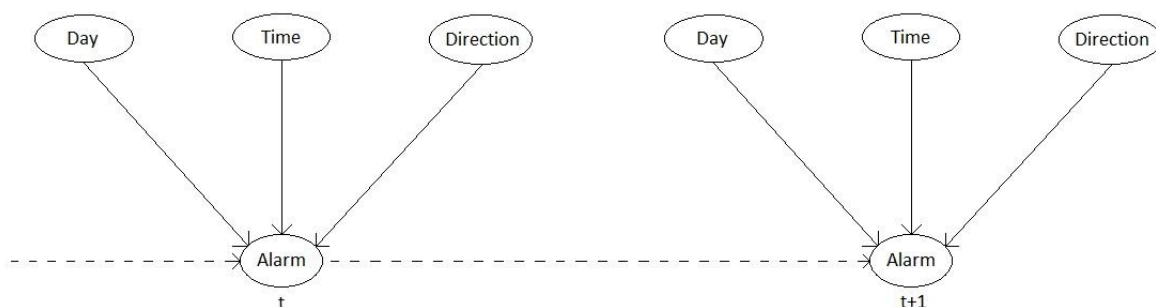


Figure 3: Dynamic Bayesian network

3.3 Variables

The variables in the Bayesian network have to represent knowledge that can be used to decide if an alarm is false or not. The registered information in the database that is available to provide this knowledge consists of vehicle-specific track-and-trace data like speed, position, and amount of fuel left in the tank. Besides this information, the date and the time the alarm occurred on are also stored. The domain experts from Allsetra have a lot of experience in filtering out and recognizing the false alarms using the available information from the database. During conversations with these domain experts, three main factors, used to make judgments, came up.

The first variable is the day of the week the alarm occurs (*day*). Because there are seven days in a week, this variable is discrete and consists of seven values. In the Allsetra database, alarms are stored with date and time in one cell. To retrieve the *day* of occurrence, first the date needs to be extracted from this cell. Then the corresponding weekday of this date is retrieved.

The second variable is the point in time the alarm occurs (*time*). In the Allsetra database, *time* is extracted from the same date/time cell as the *day*. A difference with the first variable *day*, is that *time* is continuous. To maintain a low complexity *time* is made discrete by dividing a day into six periods. This way, the variable *time* consists of six values: Midnight (12.00 AM - 03.59 AM), Early Morning (04.00 AM - 07.59 AM), Late Morning (08.00 AM - 11.59 AM), Early Afternoon (12.00 PM - 03.59 PM), Late Afternoon (04.00 PM - 07.59 PM), and Evening (08.00 PM - 11.59 PM).

The third variable is the *direction* in which the machine is moving. In the Allsetra database, for each alarm, the direction is stored in degrees. To use the direction measured at one moment would not give a very accurate estimation on where the machine is moving to, because most routes usually are not one-directional all the time. Fortunately, the box that

sends the alarm to Allsetra sends an alarm every two minutes during the movement. This way, multiple directions from multiple alarms can be averaged and a more precise direction can be generated. Another way to enhance the correctness of the direction is to take in mind that the first minutes of a route often are used to get to the main roads. These first minutes the measured directions can differ widely. However, when the main road is reached, the measured directions might differ less. The same situation occurs at the end of the route when the main road is left behind and the final destination comes close. To avoid these unreliable direction measurements, the first two directions and the last two directions are ignored. This results in a collection of measured directions that still need to be averaged to one communal direction. To take the average of a set of angles is not the same as to take the average of a set of numbers, because the angles are arranged in a circular fashion. So, if 10 is added to 355, the result is 5.

The degrees are discretized by dividing them in four directional groups: angles 0-89, angles 90-179, angles 180-269, and angles 270-359. These groups of angles correspond to the wind direction groups: Northeast, Southeast, Southwest, and Northwest. This way, the variable *direction* consists of four values.

The fourth variable is the *validity of the alarm*. It shows whether the alarm can be ignored or whether it needs to be checked; in other words, whether it is false or not. The connections between the variables are as follows. The *validity of the alarm* is the classification variable. Variables *day*, *time*, and *direction* each have their influence on the *validity of the alarm*; they are used as evidence variables in the network.

3.4 Static Bayesian network specifications

While considering what type of network to use, some specific characteristics of this domain were discovered. In this domain, with the variables *day*, *time*, and *direction*, it holds that the evidence of an incoming alarm is always complete. So, for each incoming alarm the values of *day*, *time*, and *direction* are known. In general Bayesian networks it could be possible that information for a certain variable is not available. In this case probabilities are calculated with all possible values for that variable in mind.

Another important aspect of this domain is that the symptoms of a false alarm from one vehicle are different from the symptoms of a false alarm from another vehicle. This follows from the fact that the false alarms are triggered by actions of persons who do not follow the agreed rules set up with the Service Center. The Bayesian network needs to make use of these habits and weekly patterns of rule-breaking and these cannot be retrieved from the historical alarm data of all vehicles in the database. Therefore, for each alarm, the historical alarm data of the vehicle linked to that alarm only, needs to be considered. As a consequence, a new probability distribution needs to be created for each incoming alarm.

Apart from the historical alarm data that changes for every incoming alarm, there also is a part of the network model that does not change. That is the part where the circumstances the new incoming alarm occurred under are compared with the circumstances earlier alarms occurred under. The method of comparing the new alarm with earlier alarms is for every alarm the same. To clarify this, the method will be explained in more detail.

An incoming alarm consists of all the information known about the vehicle and the circumstances the alarm occurred under. From this alarm the network retrieves the information it needs for the evidence variables *day*, *time*, and *direction*. These could for example be ‘Wednesday’, ‘Early Morning’, and ‘Southwest’. Next, the historical alarm data of the vehicle involved in this alarm is analyzed and each alarm in the historical data is narrowed down to these three variables. This could result in the following list where each line represents an alarm that occurred in the past:

Saturday	Late Afternoon	Northwest
Wednesday	Midnight	Northeast
Monday	Early Morning	Southwest
Friday	Midnight	Southwest
Wednesday	Early Morning	Northeast

Now, for each evidence variable, a probability distribution can be calculated based on this table. The more a value for a variable is present, the higher the probability that a false alarm occurs under that circumstance. Next, the probability distribution for the classification variable *validity of the alarm* is calculated. For every combination of values a probability for the alarm being false needs to be generated. The number of combinations is calculated as follows: the number of values for *day*, *time*, *direction* and *validity of the alarm* are multiplied. This results in a list of $7 \times 6 \times 4 \times 2 = 336$ combinations which looks like this:

1	Monday	Midnight	Northeast	True
2	Monday	Midnight	Northeast	False
3	Monday	Midnight	Southeast	True
4	Monday	Midnight	Southeast	False
5	Monday	Midnight	Southwest	True
...
335	Sunday	Evening	Northwest	True
336	Sunday	Evening	Northwest	False

Combinations in this list that are more likely to happen need to get a higher probability than the combinations that are less likely to happen.

As a means of eliciting the knowledge needed, i.e. the probabilities, expert knowledge is used. Each domain expert has his own means to judge whether an alarm is false or not. Therefore, the judgments of multiple experts need to be averaged to retrieve the most trustworthy probabilities. However, specifying 336 probabilities is error-prone and time-consuming. Due to conversations with those experts a unique realistic method for comparing alarms could be introduced. It can be divided in two parts. Namely, comparing one alarm with another alarm, and comparing one alarm with a whole list of other alarms. When comparing the circumstances of two alarms, one can count the number of variables that for both alarms have the same values. Because of the three variables *day*, *time*, and *direction*, two alarms in the current Bayesian network could have none, one, two, or three values in common. When two values are the same, these values could be instantiations of *day* and *time*, *day* and *direction*, or *time* and *direction*. The following list shows the possible combinations of matching circumstances for two alarms:

1	The values for <i>day</i> , <i>time</i> , and <i>direction</i> are the same.	<i>day – time – direction</i>
2	The values for <i>day</i> and <i>time</i> are the same.	<i>day – time</i>
3	The values for <i>day</i> and <i>direction</i> are the same.	<i>day – direction</i>
4	The values for <i>time</i> and <i>direction</i> are the same.	<i>time – direction</i>
5	The value for <i>day</i> is the same.	<i>day</i>
6	The value for <i>time</i> is the same.	<i>time</i>
7	The value for <i>direction</i> is the same.	<i>direction</i>
8	Not one value is the same	–

The domain experts indicated that combinations 5, 6, and 7 with only one common value could not be used to decide if an alarm was false. They simply contain too little information. When an incoming alarm only has one value in common with an earlier alarm it should be ignored and treated as if not one value corresponded (combination 8).

Apart from this one-on-one comparison between two alarms, the new incoming alarm is also compared with a whole list of earlier alarms. For each earlier alarm the values for each combination (1, 2, 3, and 4) are extracted and registered. Consider, for example, the following alarm present in the list of earlier alarms:

Wednesday	Midnight	Northeast
-----------	----------	-----------

The following values for combinations 1, 2, 3, and 4 will be registered:

1	Wednesday	Midnight	Northeast
2	Wednesday	Midnight	
3	Wednesday		Northeast
4		Midnight	Northeast

Another alarm in the list of earlier alarms could be:

Friday	Midnight	Northeast
--------	----------	-----------

For this alarm the following values for combinations 1, 2, 3, and 4 will be registered:

1	Friday	Midnight	Northeast
2	Friday	Midnight	
3	Friday		Northeast
4		Midnight	Northeast

When the registered information from both alarms is combined, it reveals that the combination ‘Midnight – Northeast’ occurred twice. Knowing this, it could be justified to increase the probability of an alarm being false when it also occurred under these circumstances.

So, after this analysis, for each combination of values, it is known how many times it occurred. This information can be used to complete the probability distribution of the variable *validity of the alarm*. However, to translate these numbers of appearances into probabilities, domain experts were consulted using a questionnaire.

3.5 Questionnaire for probabilities

Intuitively, each number of appearances of a certain combination of circumstances should get a different probability. However, the questionnaire should not be too long and it is important to avoid confusing the experts by asking too many different probabilities for too many (in their eyes) similar cases.

Therefore, the questionnaire consists of twelve cases where an incoming alarm of a vehicle is given and some earlier alarms of the same vehicle are shown. Each case represents a different level of similarity between a new alarm and its history. Earlier, the possible combinations of matching circumstances for two alarms were explained and this resulted in the following remaining relevant combinations:

1	The values for <i>day</i> , <i>time</i> , and <i>direction</i> are the same.	<i>day – time – direction</i>
2	The values for <i>day</i> and <i>time</i> are the same.	<i>day – time</i>
3	The values for <i>day</i> and <i>direction</i> are the same.	<i>day – direction</i>
4	The values for <i>time</i> and <i>direction</i> are the same.	<i>time – direction</i>
8	Not one value is the same	–

Based on these combinations the twelve cases for the questionnaire are created:

1	No history under same circumstances
2	1 alarm <i>day & time</i>
3	1 alarm <i>day & direction</i>
4	1 alarm <i>time & direction</i>
5	MIN 2 <i>day & time</i>
6	MIN 2 <i>day & direction</i>
7	MIN 2 <i>time & direction</i>
8	1 alarm <i>day, time & direction</i> AND MAX 1 <i>day & time</i> . AND MAX 1 <i>day & direction</i> AND MAX 1 <i>time & direction</i>
9	1 alarm <i>day, time & direction</i> AND MIN 2 <i>day & time</i>
10	1 alarm <i>day, time & direction</i> AND MIN 2 <i>day & direction</i>
11	1 alarm <i>day, time & direction</i> AND MIN 2 <i>time & direction</i>
12	MIN 2 <i>day, time & direction</i>

In case 1 there do not exist any alarms in the historical data that occurred under comparable circumstances as the new alarm. In case 2, 3, and 4, there exists one alarm in the historical data that has two values in common with the new alarm. In case 5, 6, and 7, there exists more than one alarm in the historical data that has two values in common with the new alarm. In case 8, one alarm exists in the historical data that occurred under the same circumstances as the new alarm. Further, at most one alarm exists for each combination of two variables. In case 9, 10, and 11, there exists one alarm in the historical data that occurred under the same circumstances. Further, for each case there exist at least two alarms that have two values in common with the new alarm. In case 12 there exist at least two alarms in the historical data that occurred under the same circumstances as the new alarm.

For each case in the questionnaire, the experts are asked to give their estimation on the probability that the new alarm is false considering its historical data. They had to mark their estimated chance on a line with two scales: a numerical and a verbal. As explained in Van der Gaag, Renooij, Witteman, Aleman, and Taal (1999), most people tend to feel more at ease with verbal probability expressions than with numbers. Later, the results of the questionnaire were translated to numbers. In Appendix A the original Dutch questionnaire and the English translation can be found (keep in mind that the cases in the questionnaire were given in a random order, which means that the case numbers in the questionnaire do not correspond with the case numbers in this chapter).

Three domain experts were asked to fill in the questionnaire. They are all very experienced in diagnosing incoming alarms and have already dealt with thousands of incoming alarms. When the questionnaire was handed out, it was explained that each case contained a new incoming alarm accompanied with its historical data of false alarms. The experts were asked to give their estimation on how probable they think the new alarm is considering the historical data. The questionnaires were filled in individually; the experts were not allowed to discuss their answers.

The obtained probabilities of the three experts are shown in table 1.

	Expert 1	Expert 2	Expert 3
Case 1	0	50	25
Case 2	85	60	50
Case 3	75	60	75
Case 4	85	60	85
Case 5	85	90	50
Case 6	75	90	75
Case 7	85	90	75
Case 8	85	60	85
Case 9	100	99	85
Case 10	100	95	85
Case 11	85	99	100
Case 12	100	99	100

Table 1: Probability ratings from the three experts for the twelve different cases.

To motivate the use of the average of the probabilities provided by the experts, the inter rater reliability between those experts is calculated. Because the rating scale is continuous, Pearson's r is the most suitable coefficient to calculate here. In table 2 the different correlations between each expert couple are shown. To retrieve an overall correlation the average of all correlations is taken. This results in an average Pearson's correlation of 0.638. Apparently, there exists a positive correlation between the ratings of the three experts and therefore, it is justified to take the average probability of the three experts for each case. In table 3 all cases are lined up with their corresponding average probabilities.

	Expert 1	Expert 2	Expert 3
Expert 1	1	0.602	0.752
Expert 2	0.602	1	0.560
Expert 3	0.752	0.560	1

Table 2: Pearson's correlations for each expert couple.

	History	Judgment
1	No history under same circumstances	0%
2	1 alarm <i>day & time</i>	65%
3	1 alarm <i>day & direction</i>	77%
4	1 alarm <i>time & direction</i>	77%
5	MIN 2 <i>day & time</i>	75%
6	MIN 2 <i>day & direction</i>	80%
7	MIN 2 <i>time & direction</i>	83%
8	1 alarm <i>day, time &, direction</i> AND MAX 1 <i>day & time</i> . AND MAX 1 <i>day & direction</i> AND MAX 1 <i>time & direction</i>	77%
9	1 alarm <i>day, time &, direction</i> AND MIN 2 <i>day & time</i>	95%
10	1 alarm <i>day, time &, direction</i> AND MIN 2 <i>day & direction</i>	93%
11	1 alarm <i>day, time &, direction</i> AND MIN 2 <i>time & direction</i>	95%
12	MIN 2 <i>day, time &, direction</i>	100%

Table 3: The average probability ratings from the three experts for the twelve different cases.

The retrieved probabilities can now be used for the probability distribution of the classification variable *validity of the alarm* for every new incoming alarm. However, the historical alarm data of each vehicle is different, and therefore each probability distribution is still different. The next example will show how these probability distributions are created.

In section 3.3 a list of possible combinations of values was generated with their corresponding number of appearances in the history of alarms of one vehicle. This number of appearances can be used to find the corresponding probability in table 3. So, when the combination ‘Midnight – Northeast’ occurred four times, the corresponding probability is 83%. To create the Conditional Probability Table (CPT) each combination of values containing ‘Midnight – Northeast’ takes over that probability of 83%. To generate the complete CPT this step is repeated for each combination of values. In table 4 an example of a small part of this CPT is shown where only the combination ‘Midnight – Northeast’ is processed.

Day	Time	Direction	P(FalseAlarm)
Monday	Midnight	Northeast	0.83
Monday	Midnight	Southeast	0
Monday	Midnight	Southwest	0
Monday	Midnight	Northwest	0
Monday	Early Morning	Northeast	0
Monday	Early Morning	Southeast	0
Monday	Early Morning	Southwest	0
Monday	Early Morning	Northwest	0
Monday	Late Morning	Northeast	0
Monday	Late Morning	Southeast	0
Monday	Late Morning	Southwest	0
Monday	Late Morning	Northwest	0
Monday	Early Afternoon	Northeast	0
Monday	Early Afternoon	Southeast	0
Monday	Early Afternoon	Southwest	0
Monday	Early Afternoon	Northwest	0
Monday	Late Afternoon	Northeast	0
Monday	Late Afternoon	Southeast	0
Monday	Late Afternoon	Southwest	0
Monday	Late Afternoon	Northwest	0
Monday	Evening	Northeast	0
Monday	Evening	Southeast	0
Monday	Evening	Southwest	0
Monday	Evening	Northwest	0
Tuesday	Midnight	Northeast	0.83
Tuesday	Midnight	Southeast	0
Tuesday	Midnight	Southwest	0
Tuesday	Midnight	Northwest	0
Tuesday	Early Morning	Northeast	0
Tuesday	Early Morning	Southeast	0
Tuesday	Early Morning	Southwest	0
Tuesday	Early Morning	Northwest	0
Tuesday	Late Morning	Northeast	0
Tuesday	Late Morning	Southeast	0

Table 4: Example of a small part of a CPT for the variable Alarm.

3.6 Prior information

Earlier in this chapter it was explained that for the chosen variables *day*, *time*, and *direction* the values are always known. To give an impression on how the Bayesian network would act when the value of a variable is unknown an example calculation will be worked out.

A new incoming alarm of a vehicle occurs under the following circumstances:

Wednesday	Evening	-
-----------	---------	---

The value for *direction* is unknown. Normally, the *validity of the alarm* would be calculated via $P(\text{alarm} \mid \text{day, time, direction})$. When all three variables *day*, *time*, and *direction* are known this value could be found in the CPT of *validity of the alarm*. When one value is unknown some extra calculations need to be done:

$$\begin{aligned} &P(\text{alarm} \mid \text{Wednesday, evening}) = \\ &P(\text{alarm} \mid \text{Wednesday, evening, northeast}) \cdot P(\text{northeast}) \\ &+ P(\text{alarm} \mid \text{Wednesday, evening, southeast}) \cdot P(\text{southeast}) \\ &+ P(\text{alarm} \mid \text{Wednesday, evening, southwest}) \cdot P(\text{southwest}) \\ &+ P(\text{alarm} \mid \text{Wednesday, evening, northwest}) \cdot P(\text{northwest}) \end{aligned}$$

When no further information would be available, the probabilities of the directions would be uniform and therefore 0.25 for each direction. However, from the history of the vehicle that triggered the alarm, prior information can be subtracted and used to adjust the probabilities of the directions. This way, the directions that occurred more often in the history of the vehicle get a higher probability than the directions that occurred less often. With this calculation, the incoming alarm that misses a value can still be diagnosed and a probability for the *validity of the alarm* can be generated.

3.7 Dynamic Bayesian network specifications

The dynamic Bayesian network in this project is an extension of the static Bayesian network. Besides the variables *day*, *time*, and *direction* the diagnosis of the validity of a new incoming alarm is also influenced by the validity of the last five alarms. The idea is that, if a genuine alarm recently occurred for the vehicle linked to the new alarm, the probability that the new alarm is false decreases. However, the validity of previous alarms is not stored. Therefore, this network needs to calculate the validity for each of the five previous alarms, and with each calculation it takes the outcome of the previous alarm into account. These five previous alarms are all from different movement situations of the same vehicle. So, if a previous alarm was two minutes ago and it was triggered by the same ride of the vehicle as the present alarm, then that previous alarm is not taken in to account.

3.8 Software

The system that delivers the judgments, provided by the network, concerning the new incoming alarms, consists of two parts. The one part is the actual Bayesian network that gives an assessment on the incoming alarm. The other part is the event handler of Allsetra that calls the Bayesian network with the essential parameters and stores the outcome of the Bayesian network. Bayesian networks are developed in programming environments which are able to work with nodes representing variables and edges representing probabilistic dependencies among these variables. The event handler already exists in the system of Allsetra and needs to be adjusted to work with the Bayesian network.

The software used to create implementations of the network is the Bayes Net Toolbox for Matlab developed by Murphy (2001). Matlab is an interactive, matrix-oriented

programming language where one does not need to worry about memory allocation or type checking. This reduces development time and keeps code short and readable. Many useful toolboxes have been created for neural networks, signal processing and image processing. In this research the Bayes Net Toolbox is used which supports many types of conditional probability distributions and inference algorithms. It also supports dynamic Bayesian networks.

3.9 Data Allsetra

The used data was provided by Allsetra. When the networks were created the testing could be done live on new incoming alarms. During two weeks 375 alarms were generated and judged by the static network. Unfortunately it was not possible to run the two networks live at the same time without interfering with the normal process of the event handler. Even running the dynamic network on its own slowed down the regular process of the event handler. This means that the two networks cannot be compared on the same set of alarms, even if we would run them after each other. Therefore, the dynamic network will be compared with the static network on a small subset of alarms.

4 Validation

The goal of this research is to reduce the number of messages being handled by the Service Center. One important constraint is that only false alarms are filtered out and all valid alarms are not. A valid alarm that is judged by the network as false could result in a stolen vehicle that would never be traced back. Therefore, the false and the valid alarms will first be analyzed separately.

Because both networks could not run live at the same time, only the static network was activated and the dynamic network was tested afterwards on a few alarms by hand. The results of the dynamic network will therefore be analyzed at the end of this chapter.

4.1 Static Bayesian network results

The static network evaluated 375 alarms in two weeks. All of these alarms were false. Figure 4 shows the number of alarms on the y-axis and the different probabilities on the x-axis. 114 alarms were judged as 100% false. Only two alarms were judged as 0% false, which means that there was no evidence that could support the conclusion that these alarms were false.

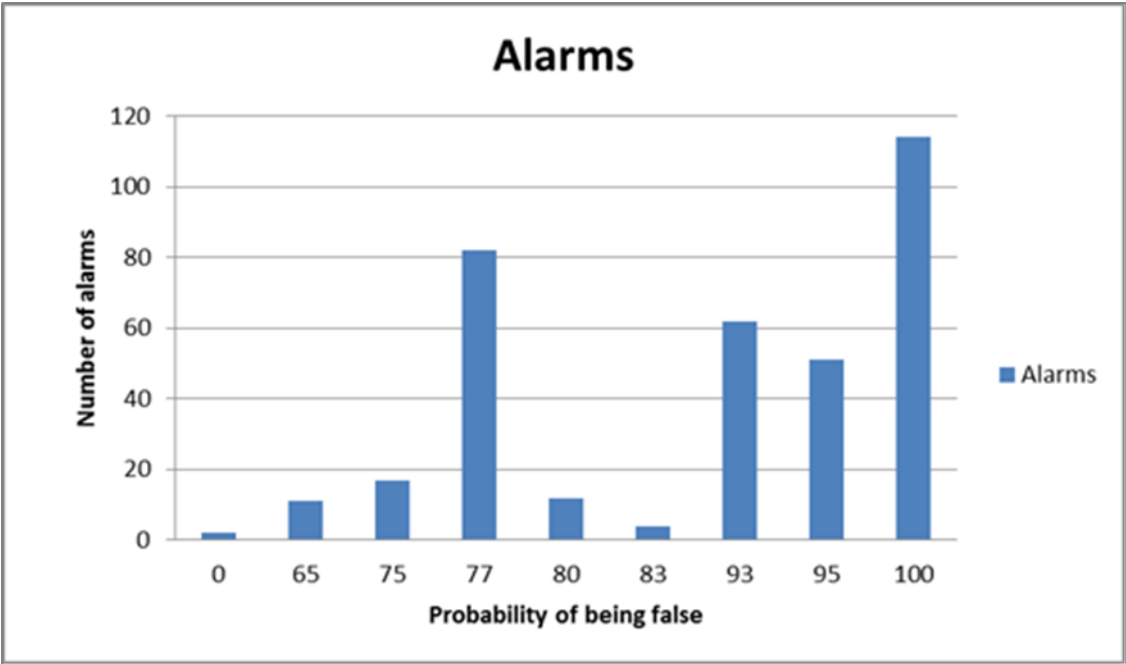


Figure 4: Diagnosis of the static network on the false alarms. On the x-axis the different probabilities for the degree of falseness are shown. On the y-axis the number of alarms is presented.

Because there were no valid alarms during the two weeks the static network was active, four valid alarms from the half year before were gathered and judged by the network. These results are shown in figure 5.

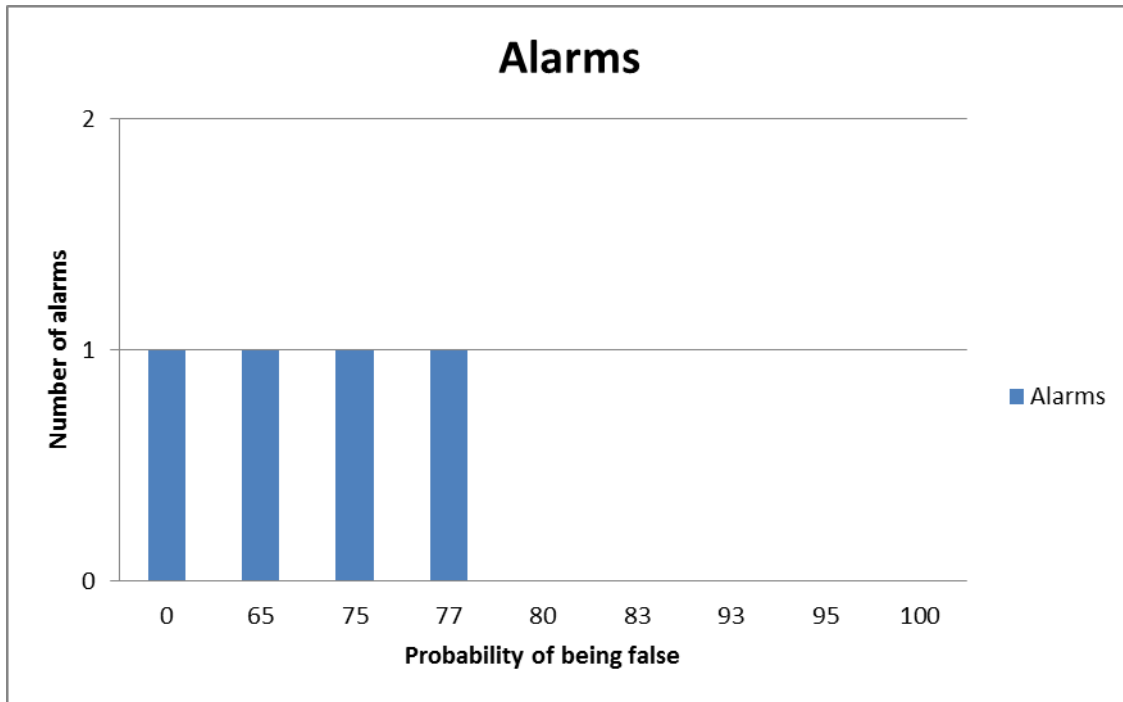


Figure 5: Diagnosis of the static network on the genuine alarms. On the x-axis the different probabilities for the degree of falseness are shown. On the y-axis the number of alarms is presented.

To make these results useful, a boundary needs to be set which defines whether the alarm is false and thus can be ignored or the alarm is valid and needs to be analyzed by the Service Center. When determining this boundary, the condition from Allsetra to avoid filtering out any valid alarm case needs to be kept in mind. Therefore, a Pareto optimal boundary needs to be found where all valid alarms are judged as valid and as many false alarms as possible are filtered out. In table 5, several situations are analyzed with different boundaries and their corresponding consequences. For each situation the number of ‘true positives’, ‘true negatives’, ‘false positives’, and ‘false negatives’ are given. The ‘true positives’ are the false alarms that are judged as false. The ‘true negatives’ are the valid alarms that are judged as valid. The ‘false positives’ are the valid alarms that are rated as false. And the ‘false negatives’ are the false alarms rated as valid.

Boundary	TP	TN	FP	FN	TP%	TN%	FP%	FN%
0%	355	0	4	0	100	0	100	0
65%	353	1	3	2	99.44	25	75	0.56
75%	342	2	2	13	96.34	50	50	3.66
77%	325	3	1	30	91.55	75	25	8.45
80%	243	4	0	112	68.45	100	0	31.55
83%	231	4	0	124	65.07	100	0	34.93
93%	227	4	0	128	63.94	100	0	36.06
95%	165	4	0	190	46.48	100	0	53.52
100%	114	4	0	241	32.11	100	0	67.89

Table 5: Each row shows the results for each boundary. TP: True Positives, TN: True Negatives, FP: False Positives, FN: False Negatives, TP%: Percentage of false alarms as TP, TN%: Percentage of valid alarms as TN, FP%: Percentage of valid alarms as FP, FN%: Percentage of false alarms as FN.

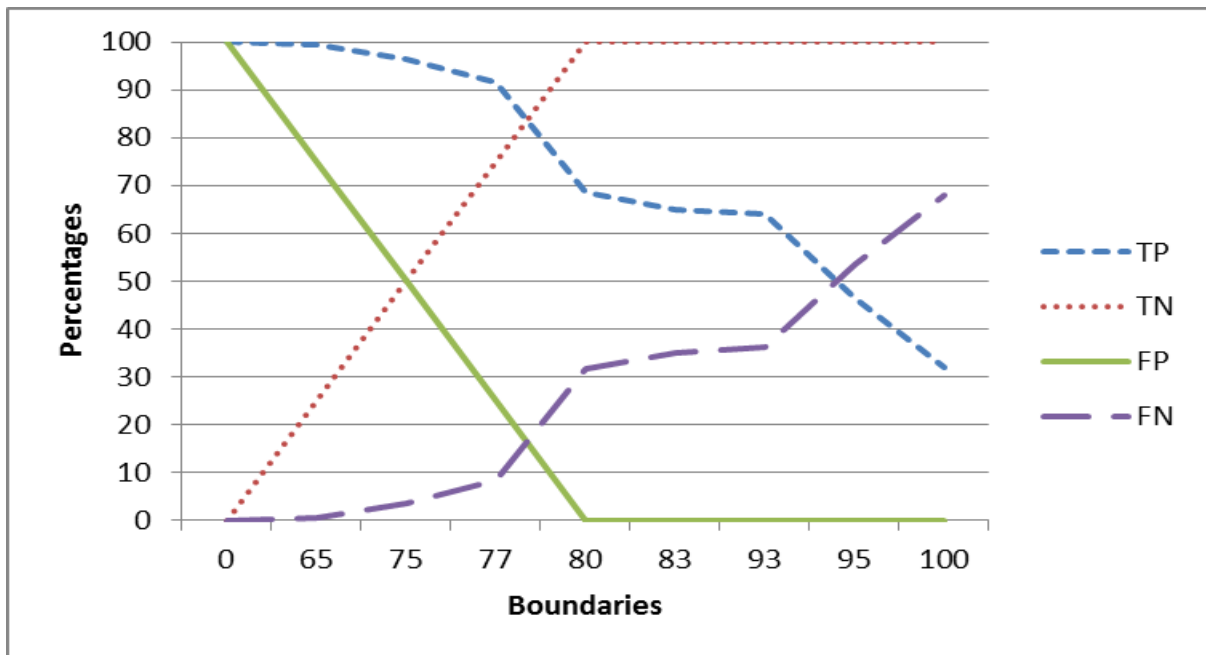


Figure 6: Representation of the results in table 4. On the x-axis the different boundaries are shown. On the y-axis the different percentages of TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives) are shown.

In figure 6 the data of table 5 is presented in a chart. With the constraint of Allsetra in mind, all valid alarms need to be rated as valid. Therefore, the Pareto optimal boundary in this situation would be 80%. 68.45% of the false alarms would then be judged as false, 100% of the valid alarms would be judged as valid.

4.2 Dynamic Bayesian network results

The dynamic Bayesian network evaluated eighteen false alarms and four genuine alarms. All these alarms were also evaluated by the static Bayesian network. It is interesting to see how the two networks compare. In table 6 the results from both networks for the eighteen false alarms are shown.

In general, the dynamic network seems to give higher probabilities. For alarm 7 and 18 considerably lower probabilities are generated. These lower probabilities could be the effect of earlier alarms that are diagnosed as genuine. The higher probabilities could be interpreted as that the dynamic network is more confident that the alarms are false due to earlier alarms that are also diagnosed as false. In table 7 the results for the genuine alarms are shown.

One probability from the dynamic network is missing, because that network at least needs the historical data of five alarms. Apparently, the vehicle where this alarm belongs to did at that moment not have enough historical data. For the alarms 2 and 4 very high probabilities were generated. Apparently, the network diagnosed earlier alarms as false, which increased the probabilities for the new alarms. Compared to the static network, the dynamic network is not very useful for recognizing the genuine alarms. With almost all possible boundaries alarms 2 and 4 would have been judged as false, which could result in two stolen and untraceable vehicles. However, the dynamic network could be useful when the probability judgments overall would be less high. At the moment, almost every judgment is over 50%. So, for the dynamic network, there is a high chance that the five previous alarms are over 50% and therefore, a lot of dynamic network judgments are too high. When the probabilities would be better divided on the 0-100 scale, the dynamic network could become more useful.

	Static probability	Dynamic probability
1	0	0
2	0	0
3	65	99.54
4	65	94.82
5	65	78.78
6	65	77.16
7	65	11.16
8	77	100
9	77	100
10	80	95.49
11	80	94.98
12	83	78.8
13	95	99.99
14	95	99.95
15	95	94.59
16	100	99.95
17	100	93.18
18	100	79.29

Table 6: Probability results from the static and the dynamic network for the eighteen false alarms.

	Static probability	Dynamic probability
1	0	-
2	65	99.33
3	75	77
4	77	99.33

Table 7: Probability results from the static and the dynamic network for the four genuine alarms.

5 Discussion

Bayesian networks have already proven useful in domains with partly comparable circumstances as the domain of theft alarm analysis. Therefore, the goal of this project was to investigate if a Bayesian network could also be used to diagnose incoming theft alarms. Practically, in the Allsetra environment, this means that the number of messages being handled by the co-workers of the Service Center need to be reduced. To achieve this, a Bayesian network was implemented and during two weeks, this network assisted in diagnosing incoming alarms. After evaluation of these diagnoses of the network the results seemed promising as more than 60% of the false alarms were diagnosed as false.

For this network to be useful, regularities in data from the past needed to be on hand to draw conclusions in the present, which was referred to in the first research question from the introduction. Three variables were found that could be modeled to fit in the network. Together these variables were able to represent returning patterns which were recognized in new incoming alarms. Due to this ability, a probability of falseness of an alarm could be created.

In the second research question it was questioned if the experts would be able to provide enough information for the Bayesian network to make correct probability judgments concerning incoming alarms. The domain experts were questioned about the most important situations, where a new alarm is compared with earlier alarms. The retrieved probabilities of the three experts were compared and it turned out that their probability rates correlated. This correlation supported the use of these probabilities by the Bayesian network to make judgments. However, whether a judgment is correct or not depends on where the boundary of genuine/false is set. The lower the boundary is set the more alarms will be judged as false and the risk of judging a genuine alarm as false grows. The higher the boundary is set the less alarms will be judged as false and the risk of judging a genuine alarm as false decreases. The position of the boundary decides how much risk is being taken.

The last research question in the introduction was if it would be possible to have a Pareto optimal working network. Allsetra wants as many false alarms as possible filtered out, but not one genuine alarm is allowed to be judged as false. A genuine alarm judged as false could have a stolen vehicle that could never be found back as a consequence. So, intuitively, Allsetra cannot filter out any alarms if the company wishes to be 100% sure that not one genuine alarm would be accidentally judged as false. In this situation, to be able to filter out any false alarms, a negligible chance that a genuine alarm is filtered out should be allowed. Keeping this in mind, interesting results were obtained. When, for the static network, the boundary of being false is set as high as possible, i.e. only 100% false is judged as false, 32% of the false alarms are filtered out. With this boundary the risk of genuine alarms judged as false is as low as possible (besides the option of handling each alarm as genuine), while still a considerable number of false alarms is filtered out. When the risk is allowed to be a bit higher, the boundary could be set on 80%. In that situation, 68% of the false alarms are filtered out.

For the dynamic network, the results were less promising. Two of the four genuine alarms were judged as 100% false, which makes it impossible to create a Pareto optimal dynamic Bayesian network. The probabilities of the dynamic network for the false alarms were, in general, a bit higher than the probabilities of the static network. This difference is probably caused by the influence of the validity outcomes of earlier alarms. Apparently, for the dynamic network, the false alarms are rated more false and the genuine alarms are also rated more false.

5.1 Allsetra

For a company like Allsetra it would be interesting to translate the results of the static network to what Allsetra gains in time and to look at how the results could improve the work

done by the employees. Every alarm that is filtered out does not need to be analyzed and the owner of the vehicle does not need to be called. The number of incoming alarms per day is unpredictable and can depend on a lot of factors. Due to the weather farmers can decide to go into the field with their agricultural vehicles in the weekends instead of on Friday. This could mean that the number of alarms on Friday is less than usual and that the number of alarms in the weekend is increased. Another example of when the usual pattern of incoming alarms can be different is when the construction industry has its holiday. In that case the number of alarms could decrease during a few weeks.

In figure 6 the numbers of alarms per day in the two weeks of testing are shown. On the Mondays a peak is registered probably due to the start of the work week. In the weekends the least alarms occurred. In these two weeks of testing the work on Monday would have benefited when 68% of the alarms would have been filtered out in advance by the Bayesian network. Less time would have been spent on checking the alarms and the daily work would not have been interrupted as many times as usual. Another advantage of decreasing the number of alarms is that the employees will be more alert when an alarm does come through.

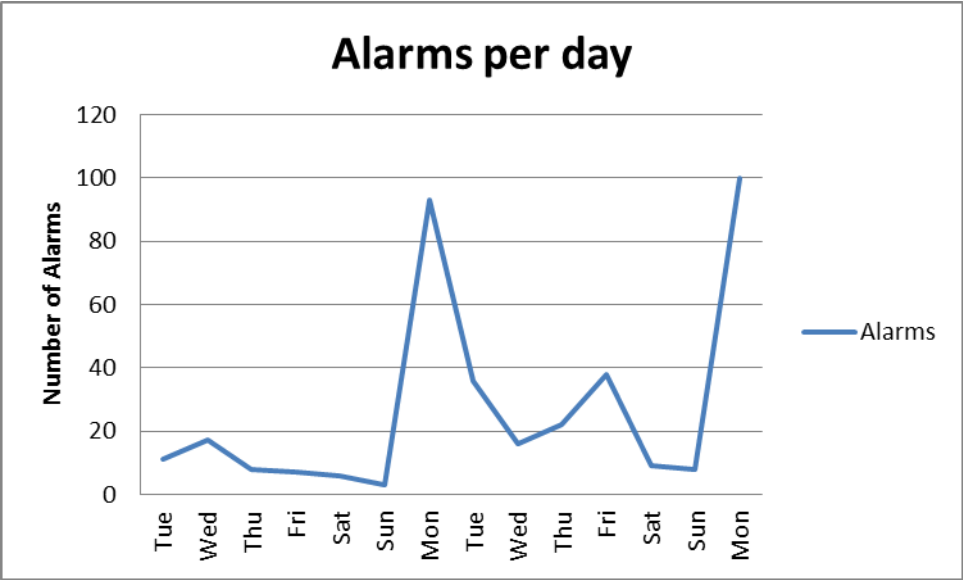


Figure 7: Number of alarms per day during the two weeks of testing.

5.2 Innovations

In this research a new method was invented to retrieve the knowledge of domain experts needed by the Bayesian network. Expert knowledge was used to create patterns that could be used to retrieve probability estimations. Due to those patterns, fewer probabilities needed to be filled in by the domain experts. Also, the estimation closely followed the way experts themselves judge the alarms.

Another new insight gained during this research is that, instead of recognizing the genuine alarms, it is more effective to recognize the false alarms. The genuine alarms do not have reoccurring patterns, while false alarms often occur under the same circumstances as earlier alarms. Therefore, the decision variable does not have the values true or false, but the values ‘support for false’ or ‘no support for false’.

5.3 Future

To improve the network and be more certain about the generated probabilities of falseness of alarms some aspects could be changed or expanded. In this project, only the three variables *day*, *time*, and *direction* are used. Other information that is stored with an alarm and can be used to subtract patterns from the historical alarm data could be added as a variable to the network. However, as a consequence of more variables, execution times for the network could increase and other running processes could be slowed down. The probabilities that were retrieved from the domain experts could also be adjusted. Doing more research on them could give more insight on how historical data can influence the validity of a new alarm. Along with this improved insight probabilities could be tweaked, which could result in more false alarms being filtered out. The dynamic network could become more useful when the probabilities would be better divided over the 0-100 scale.

5.4 Conclusion

In general, this research showed that Bayesian networks could contribute to a more efficient workspace in the domain of theft alarm analysis. This project is another example of the use of Bayesian networks in an industrial setting with promising results and proves that Bayesian networks can be suitable coworkers in an environment where uncertain information is at hand.

References

- Ben-Gal, I. (2007). Bayesian networks. In Ruggeri, F., Kenett, R., and Faltin, F., editors, *Encyclopaedia of Statistics in Quality and Reliability*. Wiley, Chichester, UK.
- Cofiño, A., Cano, R., Sordo, C., and Gutiérrez, J. (2002). Bayesian networks for probabilistic weather prediction. In *Proceedings of the 15th European Conference on Artificial Intelligence*, 695-700.
- Dey, S. and Stori, J. (2005). A Bayesian network approach to root cause diagnosis of process variations. *International Journal of Machine Tools and Manufacture*, 45(1):75–91.
- Diard, J., Bessière, P., and Mazer, E. (2003). A survey of probabilistic models, using the Bayesian Programming methodology as a unifying framework. In *Conference on Computational Intelligence, Robotics and Autonomous Systems, CIRAS*.
- Druzdzel, M. (1994). Some properties of joint probability distributions. In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, 187–194.
- Hommersom, A. and Lucas, P. (2010). Using Bayesian networks in an industrial setting: Making printing systems adaptive. In *Nineteenth European Conference on Artificial Intelligence*, 401–406.
- Kennett, R., Korb, K., and Nicholson, A. (2001). Seabreeze prediction using Bayesian networks. In *Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD 2001)*, 148–153.
- Lewis, D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *Proceedings of the European Conference on Machine Learning (ECML)*, 4-15.
- Lucas, P., De Bruijn, N., Schurink, K., and Hoepelman, A. (2000). A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU. *Artificial Intelligence in Medicine*, 3(19):251–279.
- Murphy, K. (2001). The Bayes Net Toolbox for MATLAB. *Computing Science and Statistics: Proceedings of Interface*, 33.
- Pareto, V. (1897). *Le cours d'économie politique*. Lausanne.
- Pearl, J. (1988). Probabilistic reasoning in intelligent systems: Networks of plausible inference. Morgan Kaufmann, San Mateo, CA.
- Saltelli, A. (2004). Global sensitivity analysis: An introduction. In Hanson, K. and Hemez, F., editors, *Sensitivity Analysis of Model Output*, pages 27-43. Los Alamos National Laboratory, Los Alamos, NM.
- Van Der Gaag, L., Renooij, S., Witteman, C., Aleman, B., and Taal, B. (2002). Probabilities for a probabilistic network: A case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148.

Wasyluk, H., Onisko, A., and Druzdel, M. (2001). Support of diagnosis of liver disorders based on a causal Bayesian network model. *Medical Science Monitor*, 7:327–332.

Case 3

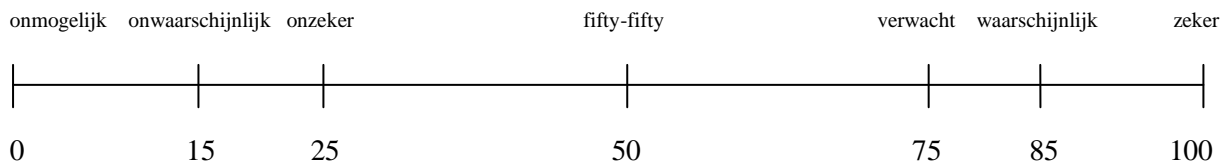
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting : ZO

Tussen de eerdere alarmen van deze box zit 1 alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Ook zijn er vier eerdere alarmen die deels overeenkomen:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Vrijdag	02.00 uur	ZO
Zaterdag	02.00 uur	ZO
Vrijdag	02.00 uur	ZO
Maandag	02.00 uur	ZO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?

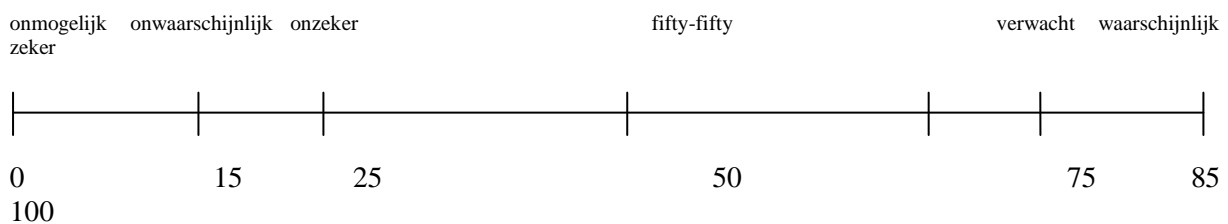


Case 4

Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit geen enkel alarm dat precies onder deze omstandigheden plaats heeft gevonden. Ook zijn er geen eerdere alarmen die voor een deel overeenkomen. Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 5

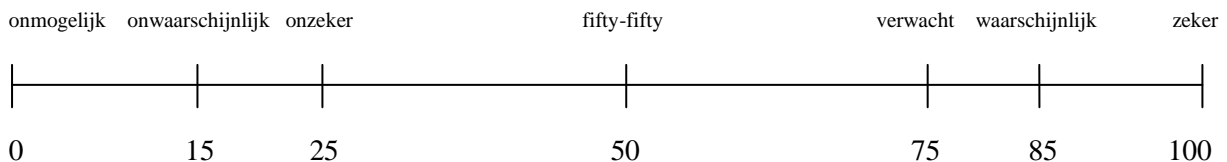
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit 1 alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Ook zijn er vier eerdere alarmen die deels overeenkomen:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Dinsdag	02.00 uur	NO
Dinsdag	02.00 uur	NW
Dinsdag	02.00 uur	ZW
Dinsdag	02.00 uur	NO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 6

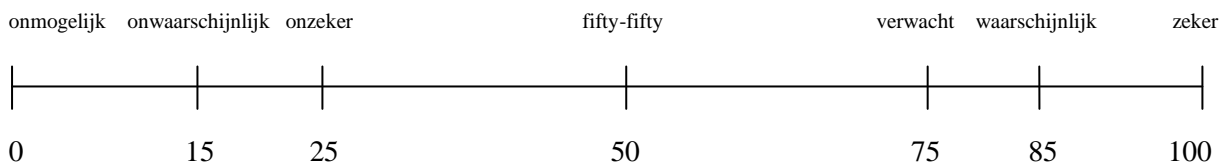
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit geen enkel alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Wel zijn er vier eerdere alarmen die deels overeenkomen:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Vrijdag	02.00 uur	ZO
Zaterdag	02.00 uur	ZO
Vrijdag	02.00 uur	ZO
Maandag	02.00 uur	ZO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 7

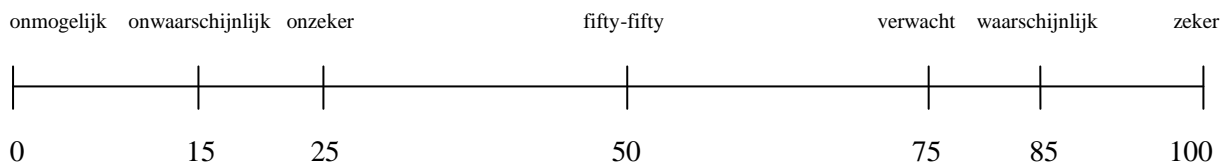
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit geen enkel alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Wel is er een eerder alarm dat deels overeenkomt:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Dinsdag	02.00 uur	NO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 8

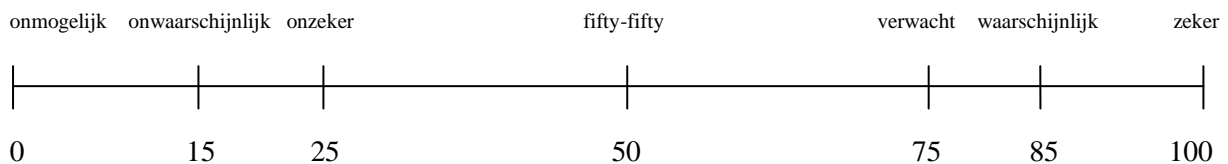
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit geen enkel alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Wel zijn er vier eerdere alarmen die deels overeenkomen:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Dinsdag	06.00 uur	ZO
Dinsdag	14.00 uur	ZO
Dinsdag	22.00 uur	ZO
Dinsdag	14.00 uur	ZO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 9

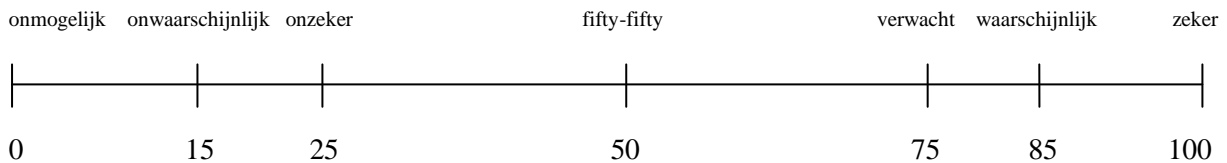
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting : ZO

Tussen de eerdere alarmen van deze box zit 1 alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Ook zijn er vier eerdere alarmen die deels overeenkomen:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Dinsdag	06.00 uur	ZO
Dinsdag	14.00 uur	ZO
Dinsdag	22.00 uur	ZO
Dinsdag	14.00 uur	ZO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?

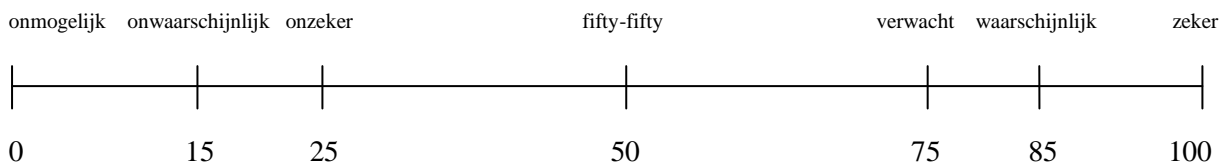


Case 10

Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit 1 alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Er zit ook maximaal 1 ander alarm tussen waar dag en tijd overeenkomen. Ook zit er maximaal 1 ander alarm tussen waar dag en richting overeenkomen. Ten slotte zit er ook maximaal 1 ander alarm tussen waar tijd en richting overeenkomen. Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?

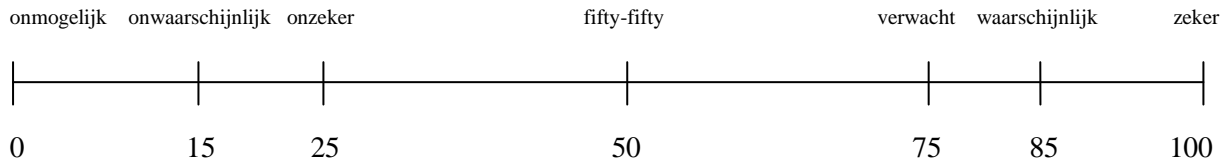


Case 11

Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zitten vier alarmen die precies onder deze omstandigheden hebben plaatsgevonden. Verder zijn er geen eerdere alarmen die voor een deel overeenkomen. Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



Case 12

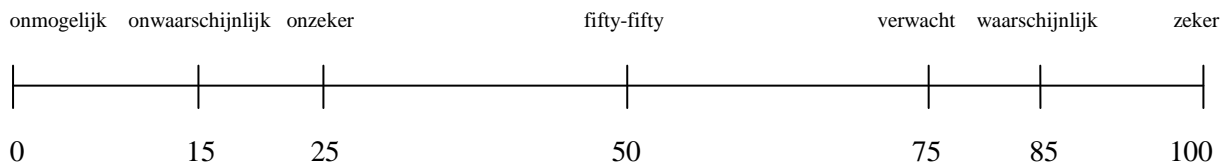
Er komt een alarm binnen waar de volgende gegevens van bekend zijn:

dag: dinsdag tijdstip: 02.00 uur richting: ZO

Tussen de eerdere alarmen van deze box zit geen enkel alarm dat precies onder deze omstandigheden heeft plaatsgevonden. Wel is er een eerder alarm dat deels overeenkomt:

<u>Dag</u>	<u>Tijd</u>	<u>Richting</u>
Vrijdag	02.00 uur	ZO

Hoe waarschijnlijk vind jij het dat dit een **vals** alarm is?



A.2 Questionnaire for domain experts translated to English

Case 1

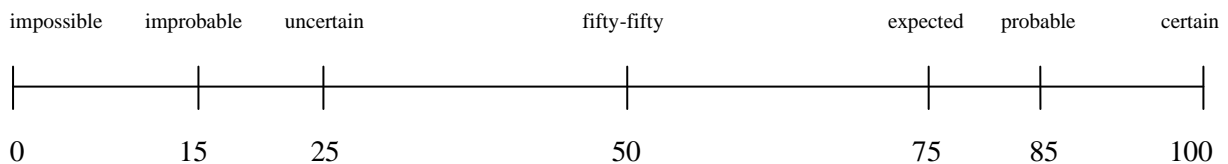
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there has been one earlier alarm that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Early Afternoon	Southeast

How probable do you think it is that this alarm is **false**?



Case 2

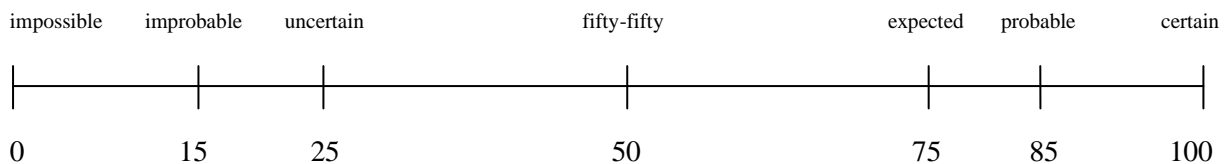
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Midnight	Northeast
Tuesday	Midnight	Northwest
Tuesday	Midnight	Southwest
Tuesday	Midnight	Northeast

How probable do you think it is that this alarm is **false**?



Case 3

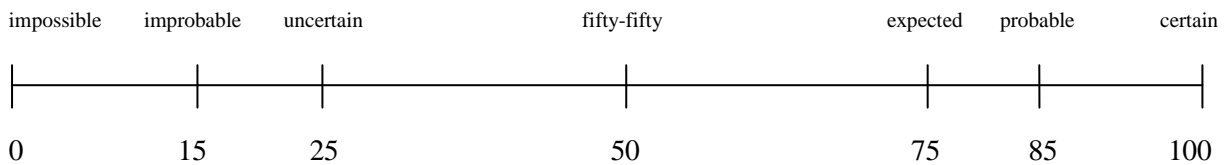
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There has been one earlier alarm for this vehicle that occurred under exactly the same circumstances. There also have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Friday	Midnight	Southeast
Saturday	Midnight	Southeast
Friday	Midnight	Southeast
Monday	Midnight	Southeast

How probable do you think it is that this alarm is **false**?

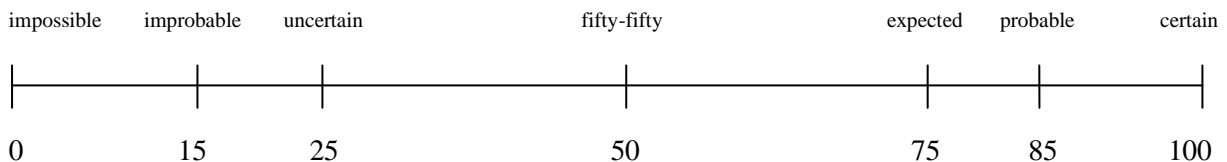


Case 4

An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. There also have not been any earlier alarms that occurred partly under the same circumstances. How probable do you think it is that this alarm is **false**?



Case 5

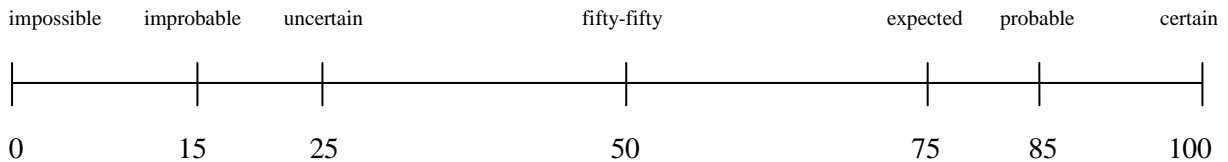
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There has been one earlier alarm for this vehicle that occurred under exactly the same circumstances. There also have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Midnight	Northeast
Tuesday	Midnight	Northwest
Tuesday	Midnight	Southwest
Tuesday	Midnight	Northeast

How probable do you think it is that this alarm is **false**?



Case 6

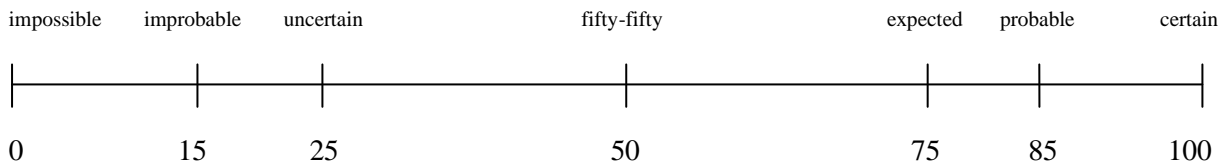
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Friday	Midnight	Southeast
Saturday	Midnight	Southeast
Friday	Midnight	Southeast
Monday	Midnight	Southeast

How probable do you think it is that this alarm is **false**?



Case 7

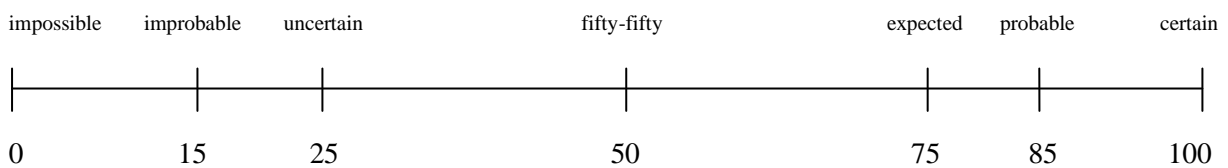
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there has been one earlier alarm that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Midnight	Northeast

How probable do you think it is that this alarm is **false**?



Case 8

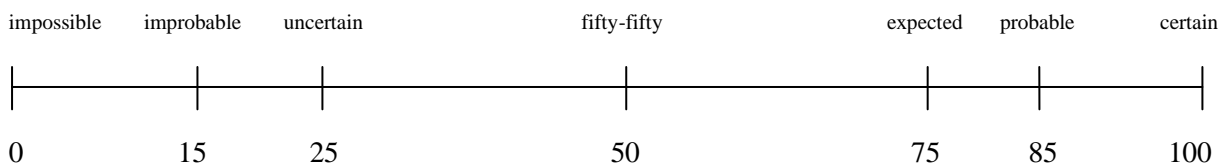
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Early Morning	Southeast
Tuesday	Early Afternoon	Southeast
Tuesday	Evening	Southeast
Tuesday	Early Afternoon	Southeast

How probable do you think it is that this alarm is **false**?



Case 9

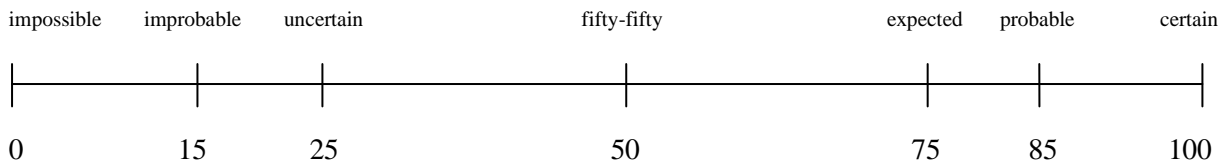
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There has been one earlier alarm for this vehicle that occurred under exactly the same circumstances. There also have been four earlier alarms that occurred partly under the same circumstances:

Day	Time	Direction
Tuesday	Early Morning	Southeast
Tuesday	Early Afternoon	Southeast
Tuesday	Evening	Southeast
Tuesday	Early Afternoon	Southeast

How probable do you think it is that this alarm is **false**?



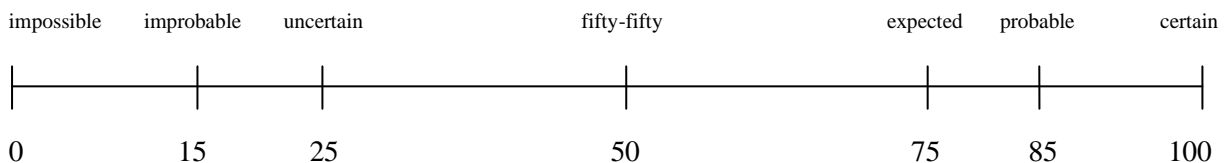
Case 10

An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There has been one earlier alarm that occurred under exactly the same circumstances. There also has been at most one alarm where day and time were the same. There also has been at most one alarm where day and direction were the same. There also has been at most one alarm where time and direction were the same.

How probable do you think it is that this alarm is **false**?



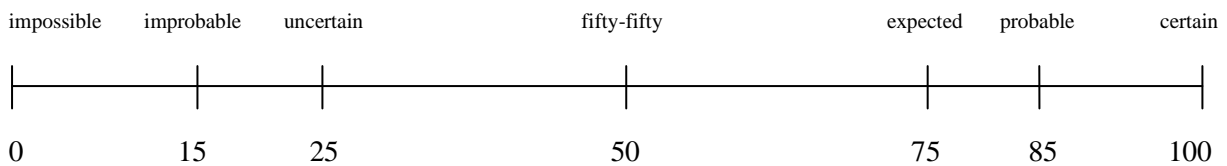
Case 11

An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been four earlier alarms for this vehicle that occurred under exactly the same circumstances. There have not been any earlier alarms that occurred partly under the same circumstances.

How probable do you think it is that this alarm is **false**?



Case 12

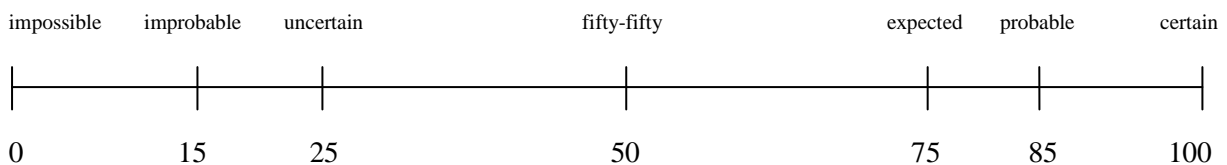
An alarm occurs under the following circumstances:

Day: Tuesday Time: Midnight Direction: Southeast

There have been no earlier alarms for this vehicle that occurred under exactly the same circumstances. However, there has been one earlier alarm that occurred partly under the same circumstances:

Day	Time	Direction
Friday	Midnight	Southeast

How probable do you think it is that this alarm is **false**?



Appendix B

Source code

B.1 Static network

statisch.m

```
function [exdata] = statisch(unitid, upId, topRecords, dbName)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:
% unitid
% upID
% topRecords
%
% Output:
% Probability --> Kans op vals alarm
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Database connection
conn = database('','*****','*****','*****',['*****']);

%Retrieve history with SQL from database
sqlquery_box_history = ['SELECT TOP ', topRecords, '          U.Id,
UP.RideId,          UPPDatumTijd.Value AS [DateTime],
REPLACE(UPPDirection.Value, ',',',', '.') AS [Direction],
REPLACE(UPPSpeed.Value, ',',',', '.') AS [Speed]          FROM          Unit AS U,
UnitPosition AS UP,          UnitPositionProtocol AS UPPAlarm,
UnitPositionProtocol AS UPPDatumTijd,          UnitPositionProtocol AS
UPPDirection,          UnitPositionProtocol AS UPPSpeed          WHERE          U.Id IN
(',',unitid,')          AND U.Id = UP.UnitId          AND UP.Id =
UPPAlarm.UnitPositionId          AND UPPAlarm.ProtocolPropertyId = 11
AND (UPPAlarm.Value LIKE '%2%' OR UPPAlarm.Value LIKE '%3%')          AND
UPPDatumTijd.UnitPositionId = UPPAlarm.UnitPositionId          AND
UPPDatumTijd.ProtocolPropertyId = 4          AND UPPDirection.UnitPositionId
= UPPAlarm.UnitPositionId          AND UPPDirection.ProtocolPropertyId = 8
AND UPPSpeed.UnitPositionId = UPPAlarm.UnitPositionId          AND
UPPSpeed.ProtocolPropertyId = 7          AND UP.Id < ',upId,'          AND
UP.RideId IS NOT NULL          AND (SELECT MDRide FROM Ride WHERE Id =
UP.RideId) = 1          AND (SELECT COUNT(*) FROM UnitPosition WHERE RideId =
UP.RideId) >= 6          ORDER BY          UPPDatumTijd.Value DESC'];
box_history = fetch(conn, sqlquery_box_history);

sqlquery_ride_history = ['SELECT          U.Id,          UP.RideId,
UPPDatumTijd.Value AS [DateTime],          REPLACE(UPPDirection.Value, ',',',',
'.'.') AS [Direction],          REPLACE(UPPSpeed.Value, ',',',', '.') AS
[Speed]          FROM          Unit AS U,          UnitPosition AS UP,
UnitPositionProtocol AS UPPAlarm,          UnitPositionProtocol AS
UPPDatumTijd,          UnitPositionProtocol AS UPPDirection,
UnitPositionProtocol AS UPPSpeed          WHERE          U.Id IN ('',unitid,')
AND U.Id = UP.UnitId          AND UP.Id = UPPAlarm.UnitPositionId          AND
UPPAlarm.ProtocolPropertyId = 11          AND (UPPAlarm.Value LIKE '%2%' OR
UPPAlarm.Value LIKE '%3%')          AND UPPDatumTijd.UnitPositionId =
UPPAlarm.UnitPositionId          AND UPPDatumTijd.ProtocolPropertyId = 4
```

```

AND UPPDirection.UnitPositionId = UPPAlarm.UnitPositionId      AND
UPPDirection.ProtocolPropertyId = 8      AND UPPSpeed.UnitPositionId =
UPPAlarm.UnitPositionId      AND UPPSpeed.ProtocolPropertyId = 7
AND UP.RideId = (SELECT DISTINCT RideId FROM UnitPosition where Id=',
upId,') ORDER BY UPPDatumTijd.Value DESC'];
ride_history = fetch(conn, sqlquery_ride_history);

% Box id + Rit id
% Haal alle alarmen op met zelfde box id en rit id

% Creëer 1 gemiddeld alarm
[dag_ev, tijd_ev, route_ev] = avgAlarms(ride_history);

% Creëer in de history voor ritten met meerdere alarmen een gemiddeld alarm
% voor die ritten.
averaged_box_history = avgHistory(box_history);

% Netwerk specs
dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
tijd_naam = {'00.00 uur tot 03.59 uur', '04.00 uur tot 07.59 uur', '08.00
uur tot 11.59 uur', '12.00 uur tot 15.59 uur', '16.00 uur tot 19.59 uur',
'20.00 uur tot 23.59 uur'};
route_opties = {'NW', 'NO', 'ZO', 'ZW'};
alarm_status = {'Alarm', 'Geen alarm'};

n=4;
dag = zeros (n,n);
d=1; t=2; r=3; a=4;
dag (d,a) = 1;
dag (t,a) = 1;
dag (r,a) = 1;
discrete_nodes = 1:n;
node_sizes = [7 6 4 2];
bnet = mk_bnet(dag, node_sizes);

% Box specs + alarm evidence
[dagen, tijden, routes, alarmen] = inlezenData(averaged_box_history);

% Informatie invoeren in netwerk
bnet.CPD{d} = tabular_CPD(bnet, d, 'CPT', dagen);
bnet.CPD{t} = tabular_CPD(bnet, t, 'CPT', tijden);
bnet.CPD{r} = tabular_CPD(bnet, r, 'CPT', routes);
bnet.CPD{a} = tabular_CPD(bnet, a, 'CPT', alarmen);

engine = jtree_inf_engine(bnet);
evidence = cell(1,n);
evidence{d} = dag_ev;
evidence{t} = tijd_ev;
evidence{r} = route_ev;

[engine, loglik] = enter_evidence(engine, evidence);

% Kans bepaald door netwerk
marg = marginal_nodes(engine, a);
vals_alarm_kans = marg.T(2)*100;

```

```
fprintf('Alarm op %s van %s richting %s. \n', dag_naam{dag_ev},
tijd_naam{tijd_ev}, route_opties{route_ev});
fprintf('Prob: %f\n', marg.T(1));
fprintf('Beoordeling:\n');
fprintf('%f procent kans dat dit een vals alarm is.\n\n', vals_alarm_kans);
```

```
% Wegschrijven
```

```
D = {dag_naam{dag_ev}, tijd_naam{tijd_ev}, route_opties{route_ev},
vals_alarm_kans};
colnames = {'dag', 'tijd', 'route', 'zekerheid vals alarm'};
exdata = D;
```

avgAlarms.m

```
function [dag_ev, tijd_ev, route_ev] = avgAlarms(ride_history)

%Datum en Tijd uit elkaar halen, richtingen opslaan in aparte array.
for i=1: length(ride_history(:,1))
    tijd_start = 0;
    length_datum = 0;
    for j=1: length(ride_history{i,3})
        if ride_history{i,3}(j) ~= ' ' && tijd_start == 0
            dagen_rij{i}(j) = ride_history{i,3}(j);
            length_datum = length_datum + 1;
        else
            tijd_start = 1;
        end
        if ride_history{i,3}(j) ~= ' ' && tijd_start == 1
            tijden_rij{i}(j-(length_datum+1)) = ride_history{i,3}(j);
        end
    end
    deg_rij(i) = ride_history(i,4);
end

%%%%%%%%%%%%% DAG %%%%%%%%%%%%%%
%Datum omzetten naar dag
for i=1: length(dagen_rij)
    dagen_rij{i} = dagDatum(dagen_rij{i});
end

for i=1: length(dagen_rij)
    dagen_rij2(i) = dagen_rij{i};
end

%Bepaal meest voorkomende dag
dag_ev = mode(dagen_rij2);

%%%%%%%%%%%%% TIJD %%%%%%%%%%%%%%
%Tijd omzetten naar tijdzone
for i=1: length(tijden_rij)
    tijden_rij{i} = tijdZone(tijden_rij{i});
end

for i=1: length(tijden_rij)
    tijden_rij2(i) = tijden_rij{i};
end

%Bepaal meest voorkomende tijdzone
tijd_ev = mode(tijden_rij2);

%%%%%%%%%%%%% RICHTING %%%%%%%%%%%%%%
%Bepaal gemiddelde richting
deg_rij = cellfun(@str2num,deg_rij);
avg_dir = averageDirection(deg_rij);

%Bepaal routenr
if (avg_dir >=0 && avg_dir < 90) || avg_dir == 360
    route_ev = 1;
else if avg_dir >= 90 && avg_dir < 180
```

```
    route_ev = 2;
else if avg_dir >= 180 && avg_dir < 270
    route_ev = 3;
else if avg_dir >= 270 && avg_dir < 360
    route_ev = 4;
end
end
end
end
```


avgHistory.m

```
function averaged_box_history = avgHistory(box_history)

rideID = box_history{1,2};
aantal_alarmen = 0;
aantal_ritten = 0;
averaged_box_history = zeros(1,3);
for i=1: length(box_history)
    if box_history{i,2} == rideID
        aantal_alarmen = aantal_alarmen + 1;
        ride_history(aantal_alarmen,:) = box_history(i,:);
    else
        aantal_ritten = aantal_ritten + 1;
        [dag_ev tijd_ev route_ev] = avgAlarms(ride_history);
        averaged_box_history(aantal_ritten,:) = [dag_ev tijd_ev route_ev];
        rideID = box_history{i,2};
        aantal_alarmen = 1;
        ride_history = {};
        ride_history(aantal_alarmen,:) = box_history(i,:);
    end
end
aantal_ritten = aantal_ritten + 1;
[dag_ev tijd_ev route_ev] = avgAlarms(ride_history);
averaged_box_history(aantal_ritten,:) = [dag_ev tijd_ev route_ev];
```

inlezenData.m

```
function [dagen, tijden, routes, alarmen_rij] =
inlezenData(averaged_box_history)

%dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
%tijd_naam = {'00.00 uur tot 03.59 uur', '04.00 uur tot 07.59 uur', '08.00
uur tot 11.59 uur', '12.00 uur tot 15.59 uur', '16.00 uur tot 19.59 uur',
'20.00 uur tot 23.59 uur'};
%route_opties = {'NW', 'NO', 'ZO', 'ZW'};
%alarm_status = {'Alarm', 'Geen alarm'};

aantal_combos = 336;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DAG VARIABLE %%%%%%%%%%%%%%%%%%%%%%%%%%

dagen_rij = averaged_box_history(:,1);
dagen_rij = dagen_rij';

%Dagen omzetten naar probabilities
dagen = zeros(1, 7);
dagen_totaal = 0;
for i=1: length(dagen)
    for j=1: length(dagen_rij)
        if dagen_rij(j) == i
            dagen(i) = dagen(i) + 1;
            dagen_totaal = dagen_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(dagen)
    dagen(i) = dagen(i)/dagen_totaal;
    if dagen(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end

%Dagen die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0
    aanpassing_niet0 = 1/dagen_totaal/(length(dagen)-nul_plekken);
    aanpassing_0 = 1/dagen_totaal/nul_plekken;

    for i=1: length(dagen)
        if dagen(i) == 0
            dagen(i) = aanpassing_0;
        else
            dagen(i) = dagen(i) - aanpassing_niet0;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIJD VARIABLE %%%%%%%%%%%%%%%%%%%%%%%%%%

tijden_rij = averaged_box_history(:,2);
tijden_rij = tijden_rij';
```

```

%Tijden omzetten naar probabilities
tijden = zeros(1, 6);
tijden_totaal = 0;
for i=1: length(tijden)
    for j=1: length(tijden_rij)
        if tijden_rij(j) == i
            tijden(i) = tijden(i) + 1;
            tijden_totaal = tijden_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(tijden)
    tijden(i) = tijden(i)/tijden_totaal;
    if tijden(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end

%Tijden die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0
    aanpassing_niet0 = 1/tijden_totaal/(length(tijden)-nul_plekken);
    aanpassing_0 = 1/tijden_totaal/nul_plekken;

    for i=1: length(tijden)
        if tijden(i) == 0
            tijden(i) = aanpassing_0;
        else
            tijden(i) = tijden(i) - aanpassing_niet0;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ROUTE VARIABLE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Richting + snelheid omzetten naar routenr
routes_rij = averaged_box_history(:,3);
routes_rij = routes_rij';

%Routes omzetten naar probabilities
routes = zeros(1,4);
routes_totaal = 0;
for i=1: length(routes)
    for j=1: length(routes_rij)
        if routes_rij(j) == i
            routes(i) = routes(i) + 1;
            routes_totaal = routes_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(routes)
    routes(i) = routes(i)/routes_totaal;
    if routes(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end

%Routes die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0

```

```

aanpassing_niet0 = 1/routes_totaal/(length(routes)-nul_plekken);
aanpassing_0 = 1/routes_totaal/nul_plekken;

for i=1: length(routes)
    if routes(i) == 0
        routes(i) = aanpassing_0;
    else
        routes(i) = routes(i) - aanpassing_niet0;
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ALARM VARIABELE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Alle probabilities op 1 gezet
alarmen_rij = ones(1, aantal_combos);

full_rij = zeros(1, aantal_combos);
part_rijDT = zeros(1, aantal_combos);
part_rijDR = zeros(1, aantal_combos);
part_rijTR = zeros(1, aantal_combos);

for i=1: length(averaged_box_history(:,1))
    %Bepaal positie waar alarm ingevoerd moet worden
    % Alarm - Richting - Tijd - Dag

    %Alledrie de waardes zijn fixed
    alarm_positie = (averaged_box_history(i,3)-1)*42; %42 mogelijke dag-
    tijd combinaties per richting
    alarm_positie = alarm_positie + ((averaged_box_history(i,2)-1)*7); %7
    mogelijke dagen per richting met tijd
    alarm_positie = alarm_positie + averaged_box_history(i,1); % per dag
    een positie verder

    full_rij(alarm_positie) = full_rij(alarm_positie) + 1;

    %Dag en tijd zijn fixed, richting is variabel

    for j=1: length(routes)
        if j ~= averaged_box_history(i,3)
            alarm_positie = (j-1)*42;
            alarm_positie = alarm_positie + ((averaged_box_history(i,2)-
1)*7);
            alarm_positie = alarm_positie + averaged_box_history(i,1);

            part_rijDT(alarm_positie) = part_rijDT(alarm_positie) + 1;
        end
    end

    %Dag en richting zijn fixed, tijd is variabel
    for j=1: length(tijden)
        if j ~= averaged_box_history(i,2)
            alarm_positie = (averaged_box_history(i,3)-1)*42;
            alarm_positie = alarm_positie + ((j-1)*7);
            alarm_positie = alarm_positie + averaged_box_history(i,1);

            part_rijDR(alarm_positie) = part_rijDR(alarm_positie) + 1;
        end
    end
end
end

```

```

    %Tijd en richting zijn fixed, dag is variabel
    for j=1: length(dagen)
        if j ~= averaged_box_history(i,1)
            alarm_positie = (averaged_box_history(i,3)-1)*42;
            alarm_positie = alarm_positie + ((averaged_box_history(i,2)-
1)*7);
            alarm_positie = alarm_positie + j;

            part_rijTR(alarm_positie) = part_rijTR(alarm_positie) + 1;
        end
    end
end

%Gebruik probabilities verkregen via experts

for i=1: length(alarmen_rij)/2
    % Zonder historie
    if full_rij(i) == 0 && part_rijDT(i) == 0 && part_rijDR(i) == 0 &&
part_rijTR(i) == 0
        alarmen_rij(i) = 1;
        fprintf('Hierzo: 1');
        % 1 66% Dag & Tijd
    else if full_rij(i) == 0 && part_rijDT(i) == 1
        if alarmen_rij(i) > (1-0.65)
            alarmen_rij(i) = (1-0.65);
            fprintf('Hierzo: 2');
        end
        % 1 66% Dag & Richting
    else if full_rij(i) == 0 && part_rijDR(i) == 1
        if alarmen_rij(i) > (1-0.77)
            alarmen_rij(i) = (1-0.77);
            fprintf('Hierzo: 3');
        end
        % 1 66% Tijd & Richting
    else if full_rij(i) == 0 && part_rijTR(i) == 1
        if alarmen_rij(i) > (1-0.77)
            alarmen_rij(i) = (1-0.77);
            fprintf('Hierzo: 4');
        end
    end
    end
end
end
end
% 1 100% <2 66%
if full_rij(i) == 1 && part_rijDT(i) < 2 && part_rijDR(i) < 2 &&
part_rijTR(i) < 2
    if alarmen_rij(i) > (1-0.77)
        alarmen_rij(i) = (1-0.77);
        fprintf('Hierzo: 5');
    end
end
end
% >=2 66% Dag & Tijd
if full_rij(i) == 0 && part_rijDT(i) >= 2
    if alarmen_rij(i) > (1-0.75)
        alarmen_rij(i) = (1-0.75);
        fprintf('Hierzo: 6');
    end
end
% >=2 66% Dag & Richting

```

```

else if full_rij(i) == 0 && part_rijDR(i) >= 2
    if alarmen_rij(i) > (1-0.80)
        alarmen_rij(i) = (1-0.80);
        fprintf('Hierzo: 7');
    end
    % >=2 66% Tijd & Richting
else if full_rij(i) == 0 && part_rijTR(i) >= 2
    if alarmen_rij(i) > (1-0.83)
        alarmen_rij(i) = (1-0.83);
        fprintf('Hierzo: 8');
    end
end
end
end
% 1 100% >=2 66% Dag & Tijd
if full_rij(i) == 1 && part_rijDT(i) >= 2
    if alarmen_rij(i) > (1-0.95)
        alarmen_rij(i) = (1-0.95);
        fprintf('Hierzo: 9');
    end
    % 1 100% >=2 66% Dag & Richting
else if full_rij(i) == 1 && part_rijDR(i) >= 2
    if alarmen_rij(i) > (1-0.93)
        alarmen_rij(i) = (1-0.93);
        fprintf('Hierzo: 10');
    end
    % 1 100% >=2 66% Tijd & Richting
else if full_rij(i) == 1 && part_rijTR(i) >= 2
    if alarmen_rij(i) > (1-0.95)
        alarmen_rij(i) = (1-0.95);
        fprintf('Hierzo: 11');
    end
end
end
end
end
% >=2 100%
if full_rij(i) >= 2
    if alarmen_rij(i) > (1-1)
        alarmen_rij(i) = (1-1);
        fprintf('Hierzo: 12');
    end
end
end
end

%Tegenovergestelde probabilities invullen
for i=1: length(alarmen_rij)/2
    alarmen_rij(i+(length(alarmen_rij)/2)) = 1-alarmen_rij(i);
end

```

dagDatum.m

```
function dag_datum = dagDatum(datum)

dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
maand_naam = {'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug',
'sep', 'oct', 'nov', 'dec'};

% 5-5-2011 23-11-2011
% 19-Dec-1994

jaar = [datum(1), datum(2), datum(3), datum(4)];
maand = [maand_naam{str2num([datum(6), datum(7)])}(1),
maand_naam{str2num([datum(6), datum(7)])}(2),
maand_naam{str2num([datum(6), datum(7)])}(3)}];
dag = [datum(9), datum(10)];

nieuwe_datum = [dag, '-', maand, '-', jaar];

[~, b] = weekday(nieuwe_datum);

for i=1: 7
    if b == dag_naam{i}
        dag_datum = i;
    end
end
```

tijdZone.m

```
function tijd_zone = tijdZone(tijd)

%tijd_naam = {'18.00 uur tot 21.59 uur', '22.00 uur tot 01.59 uur', '02.00
uur tot 05.59 uur'};
%tijd = '22:30:29';
% 7:56:33
% 12:45:13

uur = str2double([tijd(1), tijd(2)]);

if uur >= 0 && uur < 4
    tijd_zone = 1;
else if uur >= 4 && uur < 8
    tijd_zone = 2;
    else if uur >= 8 && uur < 12
        tijd_zone = 3;
        else if uur >= 12 && uur < 16
            tijd_zone = 4;
            else if uur >= 16 && uur < 20
                tijd_zone = 5;
                else if uur >= 20 && uur <= 23
                    tijd_zone = 6;
                    end
                end
            end
        end
    end
end
end
end
```


averageDirection.m

```
function avg_dir = averageDirection(degrees)

%Gebruik eerste 2 en laatste 2 richtingen niet
if length(degrees) >= 6
degrees = degrees(1,3:end-2);
end

ucomp = -1 * (sin(degrees*pi/180));
vcomp = -1 * (cos(degrees*pi/180));

if mean(ucomp) > 0
    avg_dir = (90-180/pi*atan(mean(vcomp)/mean(ucomp))+180);
elseif mean(ucomp) < 0
    avg_dir = (90-180/pi*atan(mean(vcomp)/mean(ucomp)));
elseif mean(ucomp) == 0
    if mean(vcomp) < 0
        avg_dir = 360;
    elseif mean(vcomp) > 0
        avg_dir = 180;
    else
        avg_dir = 0;
    end
end
end
```

B.2 Dynamic network

dynamisch.m

```
function [exdata] = dynamisch(unitid, upId, topRecords, dbName)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Input:
% unitId
% upId
% topRecords
%
% Output:
% Probability --> Kans op vals alarm
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Database connection
conn = database('','*****','*****','*****',['*****']);

%Retrieve history with SQL from database
sqlquery_box_history = ['SELECT      TOP ', topRecords, ' U.Id, UP.RideId,
UPPDatumTijd.Value AS [DateTime], REPLACE(UPPDirection.Value, ',', '.', '')
AS [Direction], REPLACE(UPPSpeed.Value, ',', '.', '') AS [Speed] FROM
Unit AS U,          UnitPosition AS UP,          UnitPositionProtocol AS
UPPAlarm,          UnitPositionProtocol AS UPPDatumTijd,
UnitPositionProtocol AS UPPGps,          UnitPositionProtocol AS
UPPDirection,          UnitPositionProtocol AS UPPSpeed WHERE      U.Id IN
('',unitid,')          AND U.Id = UP.UnitId          AND UP.Id =
UPPAlarm.UnitPositionId          AND UPPAlarm.ProtocolPropertyId = 11
AND (UPPAlarm.Value LIKE '%2%' OR UPPAlarm.Value LIKE '%3%')
AND UPPDatumTijd.UnitPositionId = UPPAlarm.UnitPositionId          AND
UPPDatumTijd.ProtocolPropertyId = 4          AND UPPGps.UnitPositionId =
UPPAlarm.UnitPositionId          AND UPPGps.ProtocolPropertyId = 3
AND UPPDirection.UnitPositionId = UPPAlarm.UnitPositionId          AND
UPPDirection.ProtocolPropertyId = 8          AND UPPSpeed.UnitPositionId
= UPPAlarm.UnitPositionId          AND UPPSpeed.ProtocolPropertyId = 7 AND
UP.Id < ',upId, ' AND UP.RideId IS NOT NULL AND (SELECT MDRide FROM Ride
WHERE Id = UP.RideId) = 1 AND (SELECT COUNT(*) FROM UnitPosition WHERE
RideId = UP.RideId) >= 6 ORDER BY      UPPDatumTijd.Value DESC'];
box_history = fetch(conn, sqlquery_box_history, 200);

% Creëer in de history voor ritten met meerdere alarmen een gemiddeld alarm
% voor die ritten.
averaged_box_history = avgHistory(box_history);

% Netwerk specs
dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
tijd_naam = {'06.00 uur tot 09.59 uur', '10.00 uur tot 13.59 uur', '14.00
uur tot 17.59 uur', '18.00 uur tot 21.59 uur', '22.00 uur tot 01.59 uur',
'02.00 uur tot 05.59 uur'};
route_opties = {'NW', 'NO', 'ZO', 'ZW', 'Stilstand'};
alarm_status = {'Alarm', 'Geen alarm'};

%variables
dag=1;
tijd=2;
richting=3;
```

```

alarm=4;

dag_values = 7;
tijd_values = 6;
richting_values = 5;
alarm_values = 2;

%arcs
aantal_variabelen = 4;
intra = zeros(aantal_variabelen);
inter = zeros(aantal_variabelen);

%intra-slice connections
intra(dag, alarm) = 1;
intra(tijd, alarm) = 1;
intra(richting, alarm) = 1;

%inter-slice connections
inter(alarm, alarm) = 1;

node_sizes = [dag_values tijd_values richting_values alarm_values];

%define the observed variables and the equivalence classes
obs = [dag tijd richting];
eclass1 = [dag tijd richting alarm];
eclass2 = eclass1 + aantal_variabelen;

disc = 1:aantal_variabelen;
bnet = mk_dbn(intra, inter, node_sizes, 'discrete', disc, 'eclass1',
eclass1, 'eclass2', eclass2, 'observed', obs);

%use data to retrieve probabilities for CPTs
[dag_probs, tijd_probs, richting_probs, alarm_probs, alarm_probs_next,
box_id] = retrieveHistory(averaged_box_history);

%CPTs (Conditional Probability Tables)
bnet.CPD{dag} = tabular_CPD(bnet, dag, dag_probs);
bnet.CPD{tijd} = tabular_CPD(bnet, tijd, tijd_probs);
bnet.CPD{richting} = tabular_CPD(bnet, richting, richting_probs);
bnet.CPD{alarm} = tabular_CPD(bnet, alarm, alarm_probs);

%CPTs next slice
bnet.CPD{dag + aantal_variabelen} = bnet.CPD{dag};
bnet.CPD{tijd + aantal_variabelen} = bnet.CPD{tijd};
bnet.CPD{richting + aantal_variabelen} = bnet.CPD{richting};
bnet.CPD{alarm + aantal_variabelen} = tabular_CPD(bnet, alarm +
aantal_variabelen, alarm_probs_next);

nr_slices = 5;

mat = zeros(nr_slices, 2);

%Voorgaande alarmen
observations = cell(aantal_variabelen, nr_slices);
observations = retrieveObservations(averaged_box_history, observations,
nr_slices);

```

```

%Bereken kans
for i=2:1:nr_slices
    evidence = observations(:,1:i);
    mat(i,1:2) = bereken_slice(bnet, evidence, [alarm], i);
    vals_alarm_kans = mat(i,2)*100;
end

% Wegschrijven
dag_ev = observations{1, nr_slices};
tijd_ev = observations{2, nr_slices};
route_ev = observations{3, nr_slices};

D = {box_id, dag_naam{dag_ev}, tijd_naam{tijd_ev}, route_opties{route_ev},
vals_alarm_kans};
colnames = {'box_id', 'dag', 'tijd', 'route', 'zekerheid vals alarm'};
exdata = D;
fastinsert(conn, 'Alex', colnames, exdata);

```

avgAlarms.m

```
function [dag_ev, tijd_ev, route_ev] = avgAlarms(ride_history)

%Datum en Tijd uit elkaar halen, richtingen opslaan in aparte array.
for i=1: length(ride_history(:,1))
    tijd_start = 0;
    length_datum = 0;
    for j=1: length(ride_history{i,3})
        if ride_history{i,3}(j) ~= ' ' && tijd_start == 0
            dagen_rij{i}(j) = ride_history{i,3}(j);
            length_datum = length_datum + 1;
        else
            tijd_start = 1;
        end
        if ride_history{i,3}(j) ~= ' ' && tijd_start == 1
            tijden_rij{i}(j-(length_datum+1)) = ride_history{i,3}(j);
        end
    end
    deg_rij(i) = ride_history(i,4);
end

%%%%%%%%%%%%% DAG %%%%%%%%%%%%%%
%Datum omzetten naar dag
for i=1: length(dagen_rij)
    dagen_rij{i} = dagDatum(dagen_rij{i});
end

for i=1: length(dagen_rij)
    dagen_rij2(i) = dagen_rij{i};
end

%Bepaal meest voorkomende dag
dag_ev = mode(dagen_rij2);

%%%%%%%%%%%%% TIJD %%%%%%%%%%%%%%
%Tijd omzetten naar tijdzone
for i=1: length(tijden_rij)
    tijden_rij{i} = tijdZone(tijden_rij{i});
end

for i=1: length(tijden_rij)
    tijden_rij2(i) = tijden_rij{i};
end

%Bepaal meest voorkomende tijdzone
tijd_ev = mode(tijden_rij2);

%%%%%%%%%%%%% RICHTING %%%%%%%%%%%%%%
%Bepaal gemiddelde richting
deg_rij = cellfun(@str2num,deg_rij);
avg_dir = averageDirection(deg_rij);

%Bepaal routenr
if avg_dir >=0 && avg_dir < 90
    route_ev = 1;
else if avg_dir >= 90 && avg_dir < 180
```

```
    route_ev = 2;
else if avg_dir >= 180 && avg_dir < 270
    route_ev = 3;
else if avg_dir >= 270 && avg_dir < 360
    route_ev = 4;
end
end
end
end
```

avgHistory.m

```
function averaged_box_history = avgHistory(box_history)

rideID = box_history{1,2};
aantal_alarmen = 0;
aantal_ritten = 0;
averaged_box_history = zeros(1,3);
for i=1: length(box_history)
    if box_history{i,2} == rideID
        aantal_alarmen = aantal_alarmen + 1;
        ride_history(aantal_alarmen,:) = box_history(i,:);
    else
        aantal_ritten = aantal_ritten + 1;
        [dag_ev tijd_ev route_ev] = avgAlarms(ride_history);
        averaged_box_history(aantal_ritten,:) = [dag_ev tijd_ev route_ev];
        rideID = box_history{i,2};
        aantal_alarmen = 1;
        ride_history = {};
        ride_history(aantal_alarmen,:) = box_history(i,:);
    end
end
aantal_ritten = aantal_ritten + 1;
[dag_ev tijd_ev route_ev] = avgAlarms(ride_history);
averaged_box_history(aantal_ritten,:) = [dag_ev tijd_ev route_ev];
```

retrieveHistory.m

```
function [dagen, tijden, routes, alarm_probs, alarm_probs_next] =
retrieveHistory(averaged_box_history)

%dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
%tijd_naam = {'06.00 uur tot 09.59 uur', '10.00 uur tot 13.59 uur', '14.00
uur tot 17.59 uur', '18.00 uur tot 21.59 uur', '22.00 uur tot 01.59 uur',
'02.00 uur tot 05.59 uur'};
%route_opties = {'NW', 'NO', 'ZO', 'ZW', 'Stilstand'};
%alarm_status = {'Alarm', 'Geen alarm'};

%Data inlezen

aantal_combos = 420;

%%%%%%%%%%%%% DAG VARIABLE %%%%%%%%%%%%%%

dagen_rij = averaged_box_history(:,1);
dagen_rij = dagen_rij';

%Dagen omzetten naar probabilities
dagen = zeros(1, 7);
dagen_totaal = 0;
for i=1: length(dagen)
    for j=1: length(dagen_rij)
        if dagen_rij(j) == i
            dagen(i) = dagen(i) + 1;
            dagen_totaal = dagen_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(dagen)
    dagen(i) = dagen(i)/dagen_totaal;
    if dagen(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end
end

%Dagen die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0
    aanpassing_niet0 = 1/dagen_totaal/(length(dagen)-nul_plekken);
    aanpassing_0 = 1/dagen_totaal/nul_plekken;

    for i=1: length(dagen)
        if dagen(i) == 0
            dagen(i) = aanpassing_0;
        else
            dagen(i) = dagen(i) - aanpassing_niet0;
        end
    end
end
end

%%%%%%%%%%%%% TIJD VARIABLE %%%%%%%%%%%%%%

tijden_rij = averaged_box_history(:,2);
```



```

tijden_rij = tijden_rij';

%Tijden omzetten naar probabilities
tijden = zeros(1, 6);
tijden_totaal = 0;
for i=1: length(tijden)
    for j=1: length(tijden_rij)
        if tijden_rij(j) == i
            tijden(i) = tijden(i) + 1;
            tijden_totaal = tijden_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(tijden)
    tijden(i) = tijden(i)/tijden_totaal;
    if tijden(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end

%Tijden die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0
    aanpassing_niet0 = 1/tijden_totaal/(length(tijden)-nul_plekken);
    aanpassing_0 = 1/tijden_totaal/nul_plekken;

    for i=1: length(tijden)
        if tijden(i) == 0
            tijden(i) = aanpassing_0;
        else
            tijden(i) = tijden(i) - aanpassing_niet0;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ROUTE VARIABLE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Richting + snelheid omzetten naar routenr
routes_rij = averaged_box_history(:,3);
routes_rij = routes_rij';

%Routes omzetten naar probabilities
routes = zeros(1,4);
routes_totaal = 0;
for i=1: length(routes)
    for j=1: length(routes_rij)
        if routes_rij(j) == i
            routes(i) = routes(i) + 1;
            routes_totaal = routes_totaal + 1;
        end
    end
end
nul_plekken = 0;
for i=1: length(routes)
    routes(i) = routes(i)/routes_totaal;
    if routes(i) == 0
        nul_plekken = nul_plekken + 1;
    end
end
end

```

```

%Routes die niet voorkomen in de data toch een probability geven
if nul_plekken ~= 0
    aanpassing_niet0 = 1/routes_totaal/(length(routes)-nul_plekken);
    aanpassing_0 = 1/routes_totaal/nul_plekken;

    for i=1: length(routes)
        if routes(i) == 0
            routes(i) = aanpassing_0;
        else
            routes(i) = routes(i) - aanpassing_niet0;
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ALARM VARIABELE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Alle probabilities op 1 gezet
alarmen_rij = ones(1, aantal_combos);

full_rij = zeros(1, aantal_combos);
part_rijDT = zeros(1, aantal_combos);
part_rijDR = zeros(1, aantal_combos);
part_rijTR = zeros(1, aantal_combos);

for i=1: length(averaged_box_history(:,1))
    %Bepaal positie waar alarm ingevoerd moet worden
    % Alarm - Richting - Tijd - Dag

    %Alledrie de waardes zijn fixed
    alarm_positie = (averaged_box_history(i,3)-1)*42; %42 mogelijke dag-
    tijd combinaties per richting
    alarm_positie = alarm_positie + ((averaged_box_history(i,2)-1)*7); %7
    mogelijke dagen per richting met tijd
    alarm_positie = alarm_positie + averaged_box_history(i,1); % per dag
    een positie verder

    full_rij(alarm_positie) = full_rij(alarm_positie) + 1;

    %Dag en tijd zijn fixed, richting is variabel

    for j=1: length(routes)
        if j ~= averaged_box_history(i,3)
            alarm_positie = (j-1)*42;
            alarm_positie = alarm_positie + ((averaged_box_history(i,2)-
1)*7);
            alarm_positie = alarm_positie + averaged_box_history(i,1);

            part_rijDT(alarm_positie) = part_rijDT(alarm_positie) + 1;
        end
    end

    %Dag en richting zijn fixed, tijd is variabel
    for j=1: length(tijden)
        if j ~= averaged_box_history(i,2)
            alarm_positie = (averaged_box_history(i,3)-1)*42;
            alarm_positie = alarm_positie + ((j-1)*7);
            alarm_positie = alarm_positie + averaged_box_history(i,1);

            part_rijDR(alarm_positie) = part_rijDR(alarm_positie) + 1;
        end
    end
end

```

```

end
end

%Tijd en richting zijn fixed, dag is variabel
for j=1: length(dagen)
    if j ~= averaged_box_history(i,1)
        alarm_positie = (averaged_box_history(i,3)-1)*42;
        alarm_positie = alarm_positie + ((averaged_box_history(i,2)-
1)*7);
        alarm_positie = alarm_positie + j;

        part_rijTR(alarm_positie) = part_rijTR(alarm_positie) + 1;
    end
end
end

%Gebruik probabilities verkregen via experts

for i=1: length(alarmen_rij)/2
    % Zonder historie
    if full_rij(i) == 0 && part_rijDT(i) == 0 && part_rijDR(i) == 0 &&
part_rijTR(i) == 0
        alarmen_rij(i) = 1;
        % 1 66% Dag & Tijd
    else if full_rij(i) == 0 && part_rijDT(i) == 1
        if alarmen_rij(i) > (1-0.65)
            alarmen_rij(i) = (1-0.65);
        end
        % 1 66% Dag & Richting
    else if full_rij(i) == 0 && part_rijDR(i) == 1
        if alarmen_rij(i) > (1-0.77)
            alarmen_rij(i) = (1-0.77);
        end
        % 1 66% Tijd & Richting
    else if full_rij(i) == 0 && part_rijTR(i) == 1
        if alarmen_rij(i) > (1-0.77)
            alarmen_rij(i) = (1-0.77);
        end
    end
    end
end
end
end
% 1 100% <2 66%
if full_rij(i) == 1 && part_rijDT(i) < 2 && part_rijDR(i) < 2 &&
part_rijTR(i) < 2
    if alarmen_rij(i) > (1-0.77)
        alarmen_rij(i) = (1-0.77);
    end
end
end
% >=2 66% Dag & Tijd
if full_rij(i) == 0 && part_rijDT(i) >= 2
    if alarmen_rij(i) > (1-0.75)
        alarmen_rij(i) = (1-0.75);
    end
end
% >=2 66% Dag & Richting
else if full_rij(i) == 0 && part_rijDR(i) >= 2
    if alarmen_rij(i) > (1-0.80)
        alarmen_rij(i) = (1-0.80);
    end
end
end

```

```

        % >=2 66% Tijd & Richting
    else if full_rij(i) == 0 && part_rijTR(i) >= 2
        if alarmen_rij(i) > (1-0.83)
            alarmen_rij(i) = (1-0.83);
        end
    end
end
end
end
% 1 100% >=2 66% Dag & Tijd
if full_rij(i) == 1 && part_rijDT(i) >= 2
    if alarmen_rij(i) > (1-0.95)
        alarmen_rij(i) = (1-0.95);
    end
    % 1 100% >=2 66% Dag & Richting
else if full_rij(i) == 1 && part_rijDR(i) >= 2
    if alarmen_rij(i) > (1-0.93)
        alarmen_rij(i) = (1-0.93);
    end
    % 1 100% >=2 66% Tijd & Richting
else if full_rij(i) == 1 && part_rijTR(i) >= 2
    if alarmen_rij(i) > (1-0.95)
        alarmen_rij(i) = (1-0.95);
    end
end
end
end
end
% >=2 100%
if full_rij(i) >= 2
    if alarmen_rij(i) > (1-1)
        alarmen_rij(i) = (1-1);
    end
end
end
end

%Tegenovergestelde probabilities invullen
for i=1: length(alarmen_rij)/2
    alarmen_rij(i+(length(alarmen_rij)/2)) = 1-alarmen_rij(i);
end

alarm_probs = alarmen_rij;

%%%%%%%%%%%%% ALARM NEXT VARIABELE %%%%%%%%%%%%%%

alarm_probs_next = zeros(1, length(alarm_probs)*2);

for i=1: length(alarm_probs)
    if i<((length(alarm_probs)/2)+1)
        alarm_probs_next((i*2)-1) = alarm_probs(i) + ((1-
alarm_probs(i))*0.1); % alarm_t0 = 0
        alarm_probs_next(i*2) = alarm_probs(i) - ((alarm_probs(i))*0.1); %
alarm_t0 = 1
    else if i>(length(alarm_probs)/2)
        alarm_probs_next((i*2)-1) = alarm_probs(i) -
((alarm_probs(i))*0.1); % alarm_t0 = 0
        alarm_probs_next(i*2) = alarm_probs(i) + ((1-alarm_probs(i))*0.1);
% alarm_t0 = 1
    end
end
end
end

```

retrieveObservations.m

```
function observations = retrieveObservations(averaged_box_history,  
observations, nr_slices)  
  
dag_obs = zeros(1, nr_slices);  
tijd_obs = zeros(1, nr_slices);  
richting_obs = zeros(1, nr_slices);  
  
obs_teller = nr_slices;  
for i=length(averaged_box_history(:,1)): -1 :  
length(averaged_box_history(:,1))-nr_slices+1  
    dag_obs(obs_teller) = averaged_box_history(i,1);  
    tijd_obs(obs_teller) = averaged_box_history(i,2);  
    richting_obs(obs_teller) = averaged_box_history(i,3);  
    obs_teller = obs_teller - 1;  
end  
  
observations(1,:) = num2cell(dag_obs);  
observations(2,:) = num2cell(tijd_obs);  
observations(3,:) = num2cell(richting_obs);
```

bereken_slice.m

```
function mat = bereken_slice(bnet, evidence, target, slices)

my_engine = smoother_engine(jtree_2TBN_inf_engine(bnet));
[my_engine, ll] = enter_evidence(my_engine, evidence);
marg = marginal_nodes(my_engine, [target], slices);
mat = marg.T';
```

dagDatum.m

```
function dag_datum = dagDatum(datum)

dag_naam = {'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun'};
maand_naam = {'jan', 'feb', 'mar', 'apr', 'may', 'jun', 'jul', 'aug',
'sep', 'oct', 'nov', 'dec'};

% 5-5-2011 23-11-2011
% 19-Dec-1994

jaar = [datum(1), datum(2), datum(3), datum(4)];
maand = [maand_naam{str2num([datum(6), datum(7)])}(1),
maand_naam{str2num([datum(6), datum(7)])}(2),
maand_naam{str2num([datum(6), datum(7)])}(3)}];
dag = [datum(9), datum(10)];

nieuwe_datum = [dag, '-', maand, '-', jaar];

[~, b] = weekday(nieuwe_datum);

for i=1: 7
    if b == dag_naam{i}
        dag_datum = i;
    end
end
```

tijdZone.m

```
function tijd_zone = tijdZone(tijd)

%tijd_naam = {'18.00 uur tot 21.59 uur', '22.00 uur tot 01.59 uur', '02.00
uur tot 05.59 uur'};
%tijd = '22:30:29';
% 7:56:33
% 12:45:13

uur = str2double([tijd(1), tijd(2)]);

if uur >= 0 && uur < 4
    tijd_zone = 1;
else if uur >= 4 && uur < 8
    tijd_zone = 2;
    else if uur >= 8 && uur < 12
        tijd_zone = 3;
        else if uur >= 12 && uur < 16
            tijd_zone = 4;
            else if uur >= 16 && uur < 20
                tijd_zone = 5;
                else if uur >= 20 && uur <= 23
                    tijd_zone = 6;
                    end
                end
            end
        end
    end
end
end
end
```


averageDirection.m

```
function avg_dir = averageDirection(degrees)

%Gebruik eerste 2 en laatste 2 richtingen niet
if length(degrees) >= 6
degrees = degrees(1,3:end-2);
end

ucomp = -1 * (sin(degrees*pi/180));
vcomp = -1 * (cos(degrees*pi/180));

if mean(ucomp) > 0
    avg_dir = (90-180/pi*atan(mean(vcomp)/mean(ucomp))+180);
elseif mean(ucomp) < 0
    avg_dir = (90-180/pi*atan(mean(vcomp)/mean(ucomp)));
elseif mean(ucomp) == 0
    if mean(vcomp) < 0
        avg_dir = 360;
    elseif mean(vcomp) > 0
        avg_dir = 180;
    else
        avg_dir = 0;
    end
end
end
```