Korte-termijn-voorspelling Module beschrijving

Omdat we ons verplaatsen

adviseurs mobiliteit
Goudappel Coffeng

adviseurs mobiliteit
Goudappel Coffeng

Korte termijn voorspelling
Beschrijving modules

Datum	Februari 2017
Kenmerk	
Eerste versie	

www.goudappel.nl
goudappel@goudappel.nl

Documentatiepagina

Opdrachtgever(s)	
Titel rapport	
Kenmerk	
Datum publicatie	Februari 2017

Inhoud Pagina

1

Module framework

The detection and prediction module which is developed during this project, consists of a framework of submodules and procedures. Figure 1 visualises this framework. The following chapters will describe each of the submodules in more detail. First, in chapter 2, the data-processing submodule is described. In this module, data from various data-sources is processed and fused. The fused data which is produced in the data-processing submodule is stored in the main database. In this database all relevant data is stored, both historical actual and prediction data. The database feeds other submodules and stores their results. Except for the demonstration submodule, all other submodules are implemented in the environment of OmniTRANS 8 – transport planning software developed by DAT.Mobility. In each model-run, these submodules process a set of historical data in such way that it results in detection and prediction of traffic states. Complementary, the submodules evaluate the traffic states in order to be able to detect incidents. Hereby it distincts both accidents as (regular) congestion. These submodules are described in chapter 3-6. Lastly, the demonstration module offers a traffic manager a way to visualise the results.

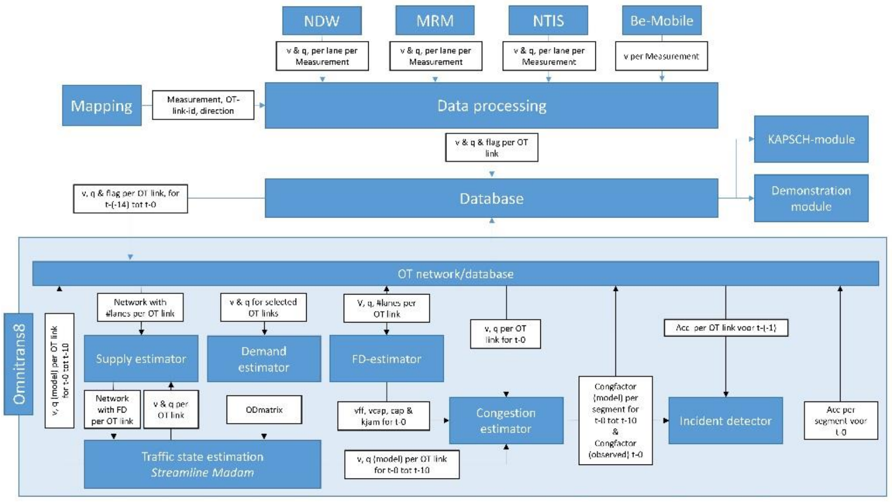

NDW
MRM
NTIS
Be-Mobile
v & q, per lane per Measurement
v & q, per lane per Measurement
v & q, per lane per Measurement
v per Measurement
Mapping
Measurement, OT-link-id, direction
Data processing
v & q & flag per OT link
KAPSCH-module
v, q & flag per OT link, for t-(-14) tot t-0
Database
Demonstration module
OT network/database
Omnitrans8
v, q (model) per OT link for t-0 tot t-10
Network with #lanes per OT link
v & q for selected OT links
V, q, #lanes per OT link
v, q per OT link for t-0
Acc per OT link voor t-(-1)
Supply estimator
Demand estimator
FD-estimator
Network with FD per OT link
v & q per OT link
ODmatrix
vff, vcap, cap & kjam for t-0
Congestion estimator
Congfactor (model) per segment for t-0 tot t-10 & Congfactor (observed) t-0
Incident detector
Acc per segment voor t-0
Traffic state estimation
Streamline Madam
v, q (model) per OT link for t-0 tot t-10

Figure 1: The framework of submodules which form the detection and prediction module

2 Data processing

2.1 Networks

In this chapter the data-processing submodules are described. Besides, this chapter is subject to a description of the most important data which is used during this project, starting with the selected networks.

The project focusses on three stretches of highway road network; One around Amsterdam (The Netherlands) and one around Birmingham (United Kingdom). The network in Amsterdam contains the A10 orbital road. For functional reasons the network includes all on- and offramps and connections to connecting highways. However, incident detection and prediction is solely performed for the A10 orbital road itself. Figure 2, visualizes the selected network.

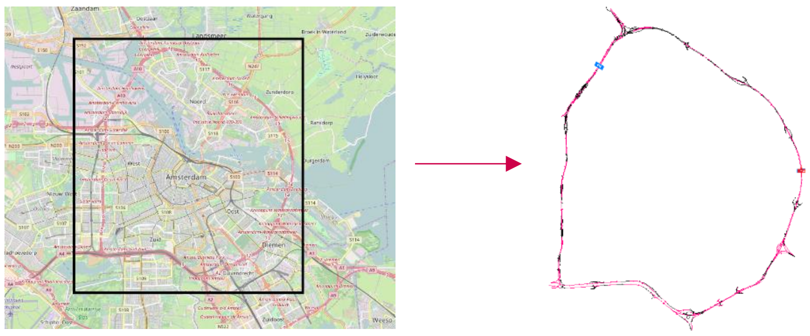


Figure 2: Visualization of the selected network for The Netherlands

The network in Birmingham contains the M42/M6/M5 orbital road. For functional reasons again the network includes all on- and offramps and connections to connecting highways. However, incident detection and prediction is solely performed for the orbital road itself. Figure 3 visualizes the selected network containing only the highway orbital road.

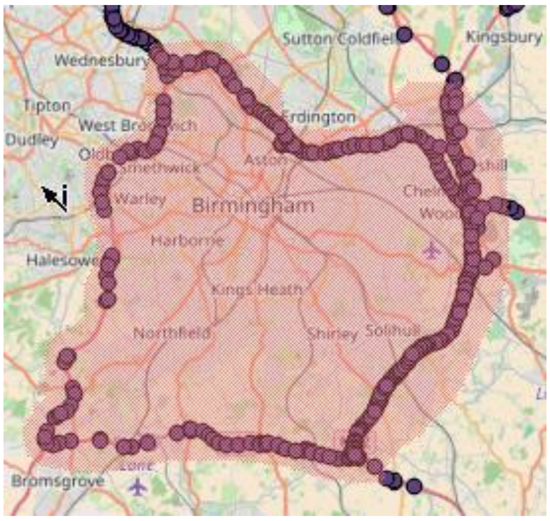


Figure 3: Visualization of the selected highway orbital network Birmingham

In Almere an important road from the city center to the highway A6 is choosen, which contains 10 junctions

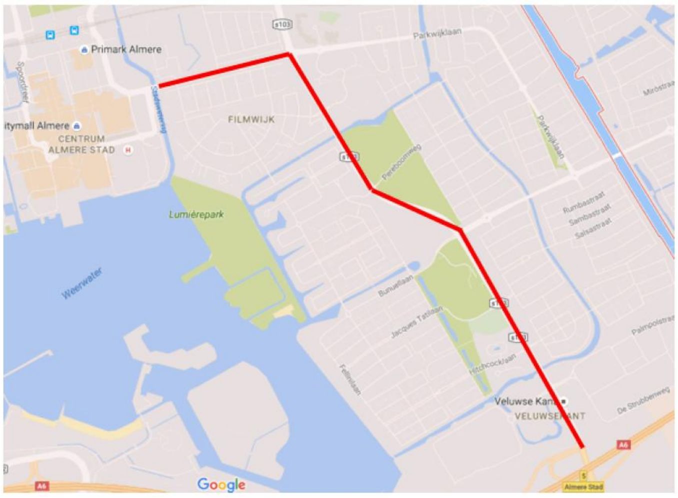


Figure 4 Selected urban road Almere

Data processing

Multiple types of submodules are distinguished within this project: data receiving submodules, mapping modules, filtering modules and fusing modules.

2.1.1 Data receiving submodules

Technically the data receiving submodules are implemented as processors in Apache NIFI. Each of the processors is described below.

NDW processor
The NDW processor actively pulls the data from the NDW website. The data (formatted in DATEX II) has a latency of about 3 à 4 minutes. The data is delivered on lane level; it contains data categorized in vehicle lengths to diverse number of cars, small trucks and long trucks. It also contains aggregated data per lane: vehicle count (flow or volume) and speed. The speed can be calculated using different algorithms.

We take the aggregated data as is and aggregate it ourselves to roadway using a weighted average to calculate the speed and sum the volumes over the links.

NTIS processor
The NTIS processor looks a lot like the NDW processor, but there are a few differences:
a) The data is not pulled, but pushed by HE.
b) The data's detail level is not categorized by car length, but in speed categories.
c) It does not contain the aggregated level per lane.

The car length categories are aggregated to a single flow and speed per lane and after that it is aggregated again to roadway level.

MRM processor
The MRM processor should consume data (pushed by NDW) and has the same level of detail as the NDW data, but should have a latency of less than a minute. However, the data feed is not available yet. Due to the delay we our now calibrating our modules without MRM data. This means we are working with a latency of 3 minutes instead of less than a minute for LDD.

Be-Mobile processor
Be-Mobile delivers speed data in CSV format, and we pull the data every minute.

VLOG-processor
VLOG data describe the status of different detection loops used for traffic lights. These data are converted to number of passing vehicles per lane per minute.

2.1.2 Mapping submodules
There is actually one mapping submodule. In this submodule it is described how to map each segment from the incoming data to one or multiple segments of the internal network which is used in the model.

2.1.3 Filtering submodules
Each data receiving submodule has a filtering module. It filters data containing invalid data (obvious errors like negative speeds and flows), but not for the detecting submodules (otherwise it would be very hard to detect incidents).

2.1.4 Fusing submodule
There is a single fusing submodule, only applied in the Amsterdam area (Birmingham has only 1 data source, so there is nothing to fuse). This submodule takes the mapped data for each segment and takes a weighted average over the speeds and flows (the weight is

dependent on the source; the current configuration weights NDW and Be-Mobile data equally important).

For datat fusion FCD en VLOG for case study Almere we refer to work package 3.

3

Traffic state estimation

For the traffic state estimation submodule it is chosen to take a model-based approach. The processed data feeds a traffic model which calculates and propagates traffic states in order to predict future traffic states. The complete traffic state estimation consists of three repetitive steps. These steps are fully executed each timestep.

Each timestep speed and flow data for the last fifteen minutes is input for this process. First the demand estimator calculates the traffic demand based on selected in-links on the network. Then the supply estimator determines the propagation conditions on the network based on actual traffic measurements and previous model settings. Given the demand and supply, the model (StreamLine::Madam) propagates traffic states on the network using macroscopic traffic flow theory. Output of this process is a set of speed and flow values per network segment for the previous 15 minutes (to be used in the supply estimator of the next run), the actual minute and for the ten minutes upcoming minutes, describing the traffic state. This chapter describes each of the three submodules concerning traffic state estimation one by one. Figure 4 illustrates the context of the state estimation.

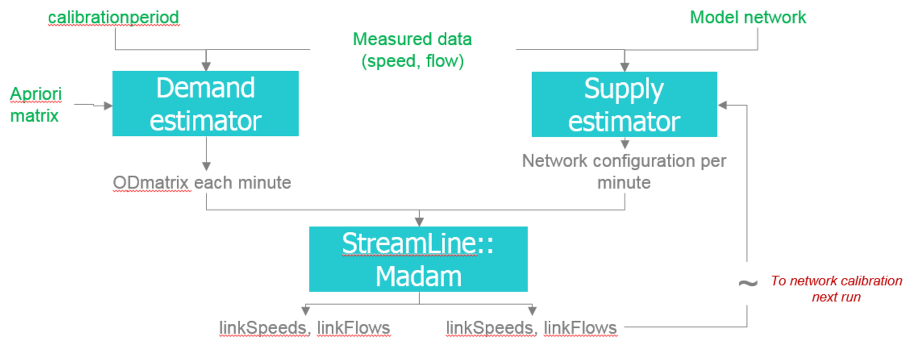


Figure 4: State estimator framework

3.1 Demand estimator

Get observed data
The traffic state estimation process starts with estimating the network demand. Therefore a set of speed and flow measurements for the last fifteen minutes is used. These speed and flow measurements are the result of the data processing submodule.

Derive demand profiles
In the network so-called inlinks are selected. These inlinks are links on the border of the chosen network and form the locations when vehicles can enter the network. This set of inlinks consists of many on- and offramps. For each of these inlinks observed data is used to determine the actual traffic demand using the latest measurement data. As flow data is not commonly available on on- and offramps, this demand is calculated using up- and downstream flow measurements. In case of congestion demand is underestimated as only actual flow is measured. Therefore, in these cases the last known demand under free flow conditions for this inlink is used for that timestep. Once traffic is in free flow conditions again, demand is calculated using up- and downstream measurement. For the prediction period, a stationary travel demand equal to the demand in the actual minute is assumed for each inlink.

Generate matrices
Once demand profiles are determined, the matrices are generated. Therefore OD pairs from an historic OD matrix are scaled in such way that traffic demand fits to the demand profiles on the inlinks. Results of this process is a set of OD-matrices (one for each minute) which can be used by the traffic model. OD-pairs that do not make use of an inlink are scaled to the average demand profile over all inlinks.

3.2 Supply estimator

Fundamental Diagram Estimator
Propagation of traffic states on the network in the model is determined by traffic flow characteristics. These characteristics are described by the fundamental diagram that is defined for each individual link. In Omnitrans the Van Aerde fundamental diagram is used. This diagram is described by four parameters: jam density (kjam), free flow speed (vff), capacity (cap) and speed at capacity (vcrit). As a result of various internal or external influences (e.g. weather conditions, speed limits, number of available lanes, amount of freight) traffic behaviour differs. This is reflected in these four parameters and so in the shape of the fundamental diagram.

Using observed data, the fundamental diagram for each link in Omnitrans is updated every minute in order to be able to reflect these pre-mentioned influences. Furthermore, updating the fundamental diagram every minute helps to be able to represent congestion on the right location on the network. Links for which no flow and speed data is available are clustered with other links up- or downstream.

The fundamental diagram is updated in three ways depending on the actual traffic conditions: Free flow, congested or around capacity. These three regimes are dynamically determined based on the Van Aerde parameters. In Figure 5 the regimes are visualized by the blue, grey and red surfaces. The boundaries of these regimes are tuning parameters.

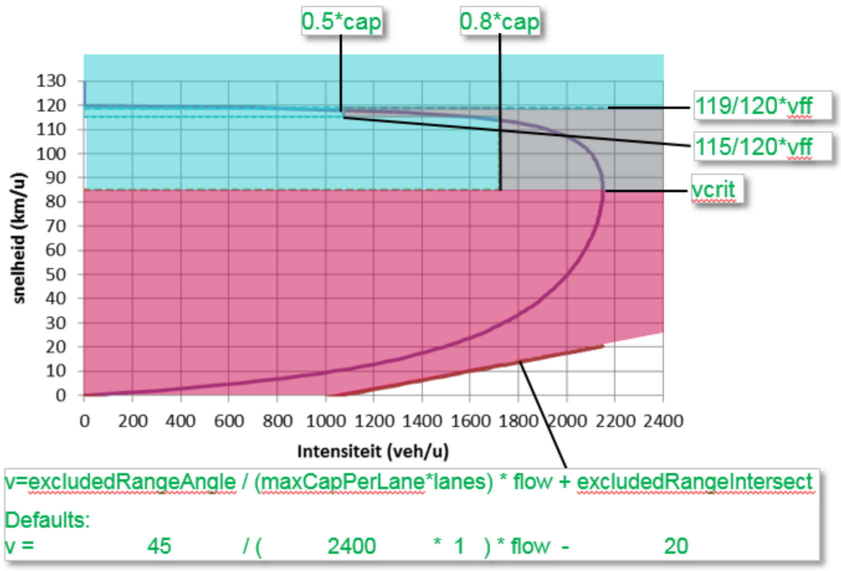
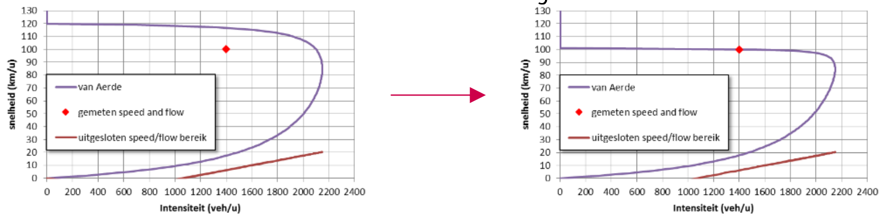
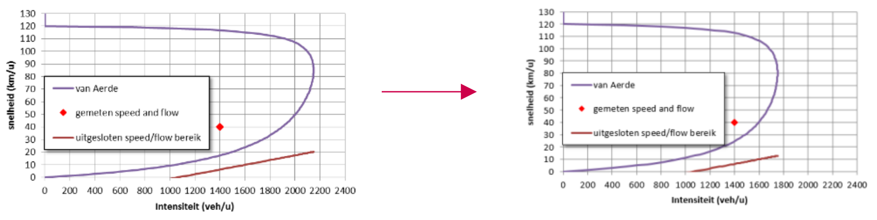


Figure 5: Visualization of the supply estimator regimes

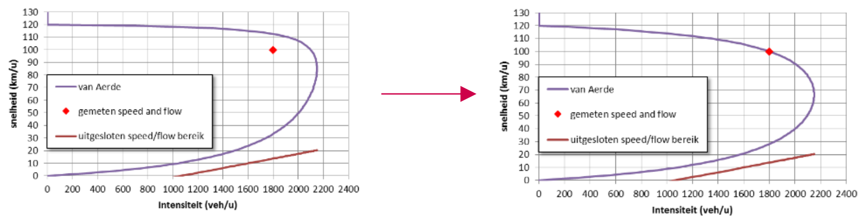
- Under free flow conditions the free flow speed parameter is adjusted in such way the the Van Aerde function describes a line through the actual traffic state.



- Under congested conditions, the capacity is adjusted proportionally to the ratio of measured speed over modelled speed in the previous run, using a rule of half while adjusting.



- Under conditions around the capacity the speed at capacity and capacity are both adjusted in such way the Van Aerde function describes a line through the actual traffic state.



3.3 Streamline::Madam

StreamLine::Madam is a macroscopic dynamic traffic assignment model that translates traffic demand on OD-level over time into traffic flows, speeds and densities on a link level for each time-period. It is consistent with traffic flow theory and described in http://www.omnitrans-international.com/nl/download/191implementation-of-a-single-dynamic-traffic-assignment-model-on-mixed-urban-and-highway-transport-networks-including-junction-modelling. Output of the Streamline::Madam submodule is the traffic state (flow, speed and density) of each segment on the selected network. The traffic state is calculated for both the actual situation as each of the timesteps in the prognosis period.

4

Congestion estimator

The congestion estimator processes both traffic states which are produced by the state estimator as actual traffic measurements. It determines the degree of congestion for both the actual situation as for each of the timesteps in the prognosis period. Outcome of the congestion estimator is a congestion factor for each network segment. The congestion factor is determined using the Van Aerde fundamental diagram. For each road segment the actual fundamental diagram (a description of actual traffic dynamics) is regularly updated. The congestion estimator points out the location of a particular prognosis traffic state or measured traffic state in this fundamental diagram. Based on its position in the fundamental diagram it can be evaluated to be free flow or congested.

This congestion factor is a continuous value between 0 and 1 which describes the degree of congestion. A congestion factor of 0 means that traffic state on the particular road segment is free flow with high certainty. A congestion factor of 1 on the other hand tells us that traffic state is congested with a high probability. For traffic states around the capacity of the road segment, traffic state measurements and prognosis are most likely to be widely scattered around the Van Aerde function. Therefore, in the congestion estimator a set of boundary conditions is set up in order to calculate a congestion factor for such traffic states between 0 and 1. The higher the congestion estimator, the more likely traffic is congested on the particular road segment. The congestion factors for all road segment are further processed in de incident detector in order to evaluate the traffic situation in more detail. The incident detector is able to identify accidents on the network and to identify the head and tail of a congestion zone.

4.1 FD estimator

In the FD-estimator an estimation for the actual form of the fundamental diagram is made. This process is fed by historical data and FD parameters for each link are regularly updated. The FD estimator can to include external effects on traffic flow dynamics such as seasonal effects.

4.2 Congestion estimator

The congestion estimator contains of a set of mathematical rules which process traffic states into congestion factors. These congestion factors are then used by the incident detector. The congestion estimator is able to process any traffic state (set of speed & flow) for road segments for which a set Van Aerde parameters is available. This set of parameters is output of the FD estimator submodule.

In this project both the output of the state estimator as actual measurements are processed through the congestion estimator. This means that the modelled traffic state for both the actual situation (timestep=0) and the full prognosis period (timestep 1-10) is processed in the congestion estimator. Next to the modelled output, also measured traffic states for timestep=0 are processed in the congestion estimator. This last step can only be executed if for a road segment both a flow as a speed measurement are available.

4.2.1 Set of mathematical rules

In Figure 6, an example of the functioning of the congestion estimator is visualized. Based on this example, the congestion estimator is described more extensively.

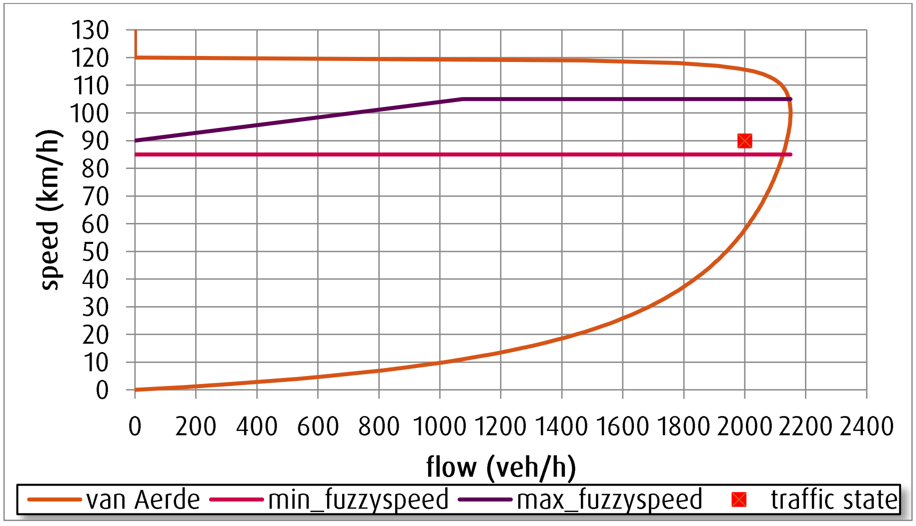


Figure 6: Example of the functionality of the congestion estimator

In this example the Van Aerde fundamental diagram is visualized for a particular road segment. This fundamental diagram is the result of the FD estimator. Based on this Van Aerde set of parameters and four additional input variables (deltamin, deltaplus, LowFlowVCRatio & LowFlowSpeed) the min_fuzzyspeed and max_fuzzyspeed boundaries are calculated. These boundaries are also visualized in Figure 6. Any traffic state with a speed below the min_fuzzyspeed boundary is said to be congested with a high probability. This traffic state is therefore processed into a congestion factor of 1. On the other hand, any traffic state with a speed which is higher than the max_fuzzyspeed boundary is said to be free flow with a high probability. This traffic state is therefore processed into a congestion factor of 0. As the max_fuzzyspeed boundary is not constant for any flow value, flow is required for congestion estimation too. At last, any traffic state which is situated between the min_fuzzyspeed and max_fuzzyspeed boundaries is scored a value between 0 and 1 depending on their location in this regime. For example, in Figure 6, the road segment with a modelled traffic state of [speed=90 km/h & flow = 2000 veh/h], is scored a congestion factor of 0,75.

5

Incident detector

In the incident detector submodule output from the congestion estimator submodule is further processed. The incident detector is split in two parts. Each of these parts focusses on another type of incident: accidents vs. congestion. First, the incident detector evaluates congestion factors for timestep=0 which are calculated for both modelled and measured traffic states in order to detect accidents. Furthermore, all modelled congestion factors are evaluated over space and time in order to locate congestion zones over the network. This way the location and movements of congestion on the network is determined.

5.1 Accident detection

The accident detection is performed for any road segment for which a congestion factor for both modelled as measured traffic states is determined. As the model produces traffic states for any road segment for each time step, this depends on the availability of a congestion factor based on actual measurements. As both flow and speed are required to calculate the congestion factor, the accidents detection is only executed for road segments with both flow and speed measurement. For this project this means that this is only done for road segments containing loop detectors.

For each road segment fulfilling the pre-described condition, the likelihood of an accidents is calculated. This is done using a set of mathematical rules with a kind of fuzziness included. If congestion is not expected by the model and traffic measurements show that there is congestion anyway, it is presumable that congestion is caused by an accident. First, the modelled congestion factors of a particular road segment are aggregated over the full prognosis period. This step is included as it is undesirable that the module detects accidents while measured congestion was in fact expected by the model a few minutes late. In this case measured congestion is still expected (although not at the correct time) and should not be marked as an accident. The aggregation is performed using a weighing principle in which the weight of each new minute in the prognosis period decreases linearly over time, yielding the following series when applied to a ten minute period. Due to findings during the tuning process this aggregation can be subject to changes later during the project.

*Aggregated_model_congestion_factor = congfactor(t0)*11/66 + congfactor(t1)*10/66 + congfactor(t2)*9/66 + ... + congfactor(t10)*1/66*

Once the aggregated modelled congestion factor is calculated it can be compared with the measured congestion factor for the same road segment. The results of this comparison is an accident factor (acc) between 0 and 1. And acc-value of 0 means that the probability that an accidents has happened is zero. An acc-value of 1 means an accident must has happened for sure. A acc-value in between 0 and 1 describes the posibility of an accident. The mathematical rules to calculate the accident factor for each road segment are described below. The rules are visualized in Figure 7.

The boundary conditions of the fuzzy regime are described by function f1 and f2 in which alfa1, alfa2, beta1 and beta2 are input variables on which the submodule can be tuned.

- f1=alfa1*x-beta1 {min=0, max=1}
- f2=alfa2*x-beta2 {min=0, max=1}

With these boundary conditions known, the observed congestion factor and de aggregated modelled congestion factor are compared with eachother using a set of logic rules:

If cong_mod > f1(cong_obs) → acc=0
Else if Cong_mod<f2(cong_obs) → acc=cong_obs
*Else acc=((f1(cong_obs)-cong_mod)/(f1(cong(obs)) f2(cong_obs)))*cong_obs*

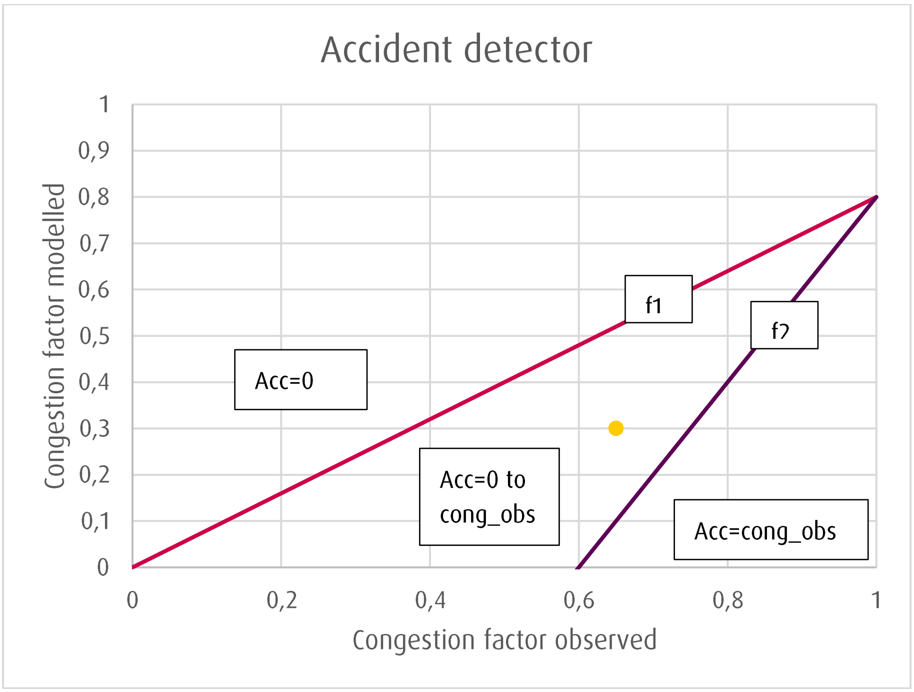


Figure 7: Visualization of the accident detector

After the actual accident factor for each road segment is calculated, the accident factor is compared with results for one timestep ago. If modelled output does not match measured congestion factor for multiple timesteps it is more likely that in fact an accident has happened. Therefore the accident factor is slightly increased to take this temporal aspect into account. The following logic rule is used to embed this phenomenon (with acc_tresh1 & acc_tresh2 to be the inputvariables for tuning the submodule).

If acc(t-1)>0 & acc(t0)-acc(t-1)>acc_tresh1 → acc=acc+acc_tresh2

6

Demonstration module

In the demonstration module - a webbased application with a graphical user interface, traffic states are visualized on a map. The demonstration module visualizes traffic state predictions as well as actual and historical traffic states. Besides traffic states, accidents are also visualized on the map. A screenshot of the demonstration module for Amsterdam is shown in figure 8.

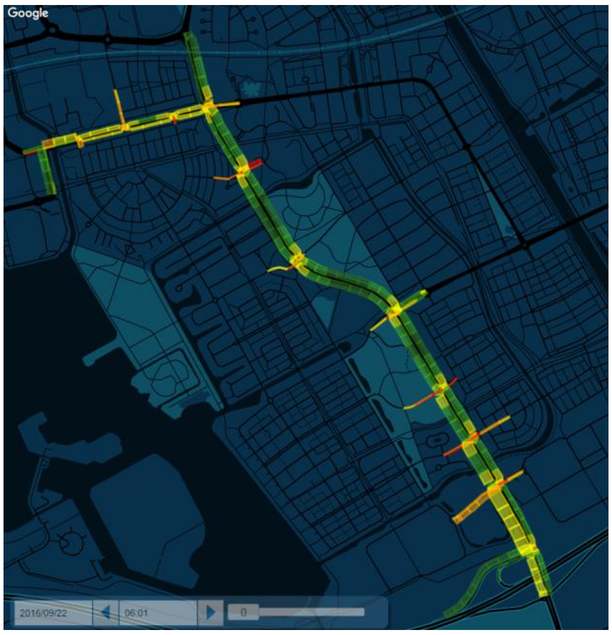


Figure 8: Preview of the demonstration module

Vestiging Deventer
Snipperlingsdijk 4
7417 BJ Deventer
T +31 (0570) 666 222
F +31 (0570) 666 888
Postbus 161
7400 AD Deventer

www.goudappel.nl
goudappel@goudappel.nl

adviseurs
mobiliteit
Goudappel
Coffeng