Incident prediction with neural networks
Onderzoek Predictive Analytics IM

A project for:

Rijkswaterstaat
Ministerie van Verkeer en Waterstaat

OPDRACHT 25-04-2016

Contents

1 BACKGROUND

2 INCIDENT PREDICTIONS WITH NEURAL NETWORKS: PREVIOUS WORK (M0)

3 DESIGN OPTIONS

4 DATA PREPARATION AND SYSTEM SETUP

5 TESTS AND RESULTS

6 CONCLUSIONS AND RECOMMENDATIONS

A APPENDIX

1

BACKGROUND

Project background

The large number of road users on the crowded Dutch roads leads to significant congestion and incidents. On an annual basis there are more than **100,000** incidents, approximately **270** incidents occur every day.

In addition to the human toll, road incidents cause severe disruptions to the circulation on the road network, with economic and social consequences.

Increasing the visibility on risk sections of the road highway, or on circumstances that may lead to incidents, is of general interest for incident management. Increasing the ability to predict when and where incidents will happen would have a major influence on our ability of mitigating the causes of incidents and, when this fails, respond appropriately.

Given the multitude of factors that lead to incidents this may well be impossible. However, improving the ability of predicting, even marginally, could be of significant value. The increasing availability of data (incidents data, traffic, weather, other data) opens the possibility of exploring methods of big data analysis for incident prediction.

This project explores a set of these methods, namely artificial neural networks, to inform RWS on the opportunity of establishing a more structural effort on incident prediction based on Big Data and Data Science.

Rationale

Traffic flow can be affected by a multitude of events, ranging from minor local disturbances (an object on the road) to major incidents blocking the entire traffic flow for hours. We call ”incident” any travel flow disturbance that leads to an action by road authorities to regulate and/or re-establish the flow.

Some of these incidents are the result of random external causes (a branch of a tree falling on the road). Many others are influenced by known factors (speed, vehicle density, weather etc.) although the influence chain appears as complex and highly non-linear.

Historical data suggests that, as a minimum, it is possible to define a baseline probability for incidents at a certain time and at a certain place. Current road management practices use this baseline augmented by human interpretation of the current traffic and travel context conditions to take action.

While it is probably impossible to precisely predict incidents for a specific time and place, it is realistic to expect that the interpretation of patters of the factors that influence incidents can lead to a higher accuracy of prediction compared to historical data alone.

The rationale for looking at incident prediction is twofold:

- To provide foresights that lead to better resource allocation
- To provide foresights that lead to more effective incident prevention measures

Incidents

Incidents include a wide range of events that may impact the traffic flow. In the practice of road inspection and incident management, incidents are recorded in classes, as follows:

- Vehicle problems (*Gestrand Voertuig*): Incidents associated to a car breakdown such as engine failure, flat tire, etc.
- Accidents (ongevallen): accidents involving one or more vehicles, such as cars, busses, trucks, leading to material damage, injuries or fatalities
- Objects on the road: incidents related to the presence of objects, such as wood, rubber or other objects.
- Other: other circumstance leading to a traffic impact, ranging from oil slick, to animals or temporary road activities.

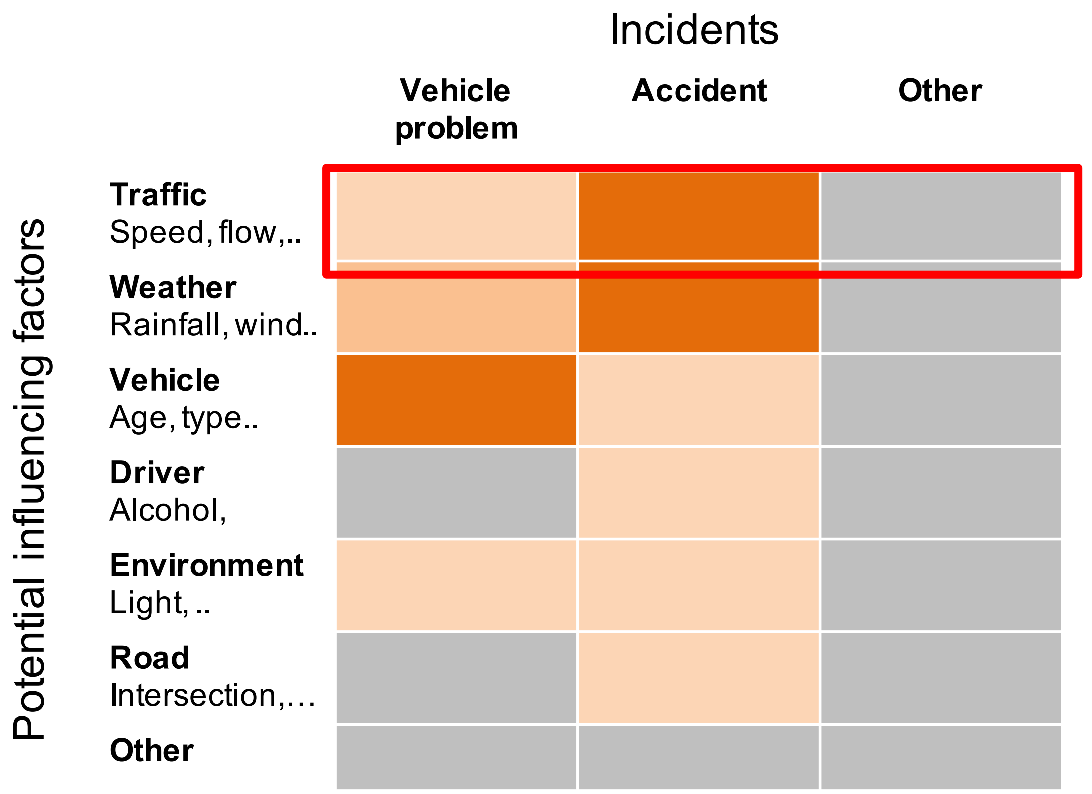
For simplicity, we group the last two classes into “Other incidents”.

Although incident registration is systematic and detailed, the way incidents are specified and classified appears as incoherent between different years, and the details include a number of redundancies and duplicate classes.

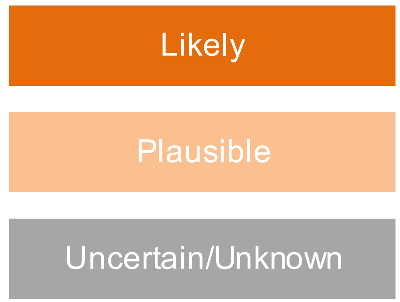
We take the classification provided by Rijkswaterstaat at “face value” and do not consider, for instance, possible double counting (a vehicle problem that leads to an accident) nor possible imprecisions in the registration of the time and location of the incident.

Incident and influencing factors

The table shows a qualitative mapping between influencing factors and incidents. For many incident types, literature and expert knowledge indicates that some influencing factors are likely or plausible. In principle, if there is a likely or plausible relationship between an influencing factor and an incident type, then it is reasonable to assume that it is at least theoretically possible to predict incidents by understanding patterns in the influencing factors.



Qualitative relationship between influencing factors and incidents



Focus on road accident: data dimensions

Influencing Factors	Data dimension (examples)
Traffic	• Speed • Flow • Congestion
Weather	• Rain • Wind • Ice on the road
Vehicle	• Age • Maintenance • Automatic safety systems
Driver	• Alcohol consumption • Experience • Seatbelt, helmet, driving style
Environment	• Light direction • Visibility • Illumination
Road	• Road layout (e.g. lanes, curvature) • Infrastructure quality • Intersections

The causes of accidents are multiple and interrelated (traffic may depend on driver behaviour, which may depend on the environment, leading to a different choice of speed), making explicit models extremely hard to design.

The alternative strategy is to train a machine to detect incidents from incident factors. If there is a sufficient history of incidents and of candidate factors, it is conceivable that a machine can learn the patterns that lead to incidents.

Addressing the prediction challenge

Within the broad range of incidents, it is reasonable to address the challenge in concentric sets, asking the following questions:

- Is it possible/meaningful to predict (better than historical averages) any type of incident?
- Is it possible/meaningful to predict (better than historical averages) specific classes of incidents that have a known qualitative dependence on traffic and context factors?
- Is it possible/meaningful to predict (better than historical averages) the specific accidents that lead to material damages and/or casualties?

We have two main strategies to address the challenge:

- Define an explicit **model** of incidents based on the influencing factors for which there is data available and optimize the model based on past evidence of incidents
- Use **machine learning** to learn - from past data - the relationship between influencing factors and incidents, without assuming a model a priori

Given the know modeling difficulties described in literature and the recent development of machine learning tools, there is a potential opportunity for address the questions above through machine learning approaches, specifically Neural Networks (NNs).

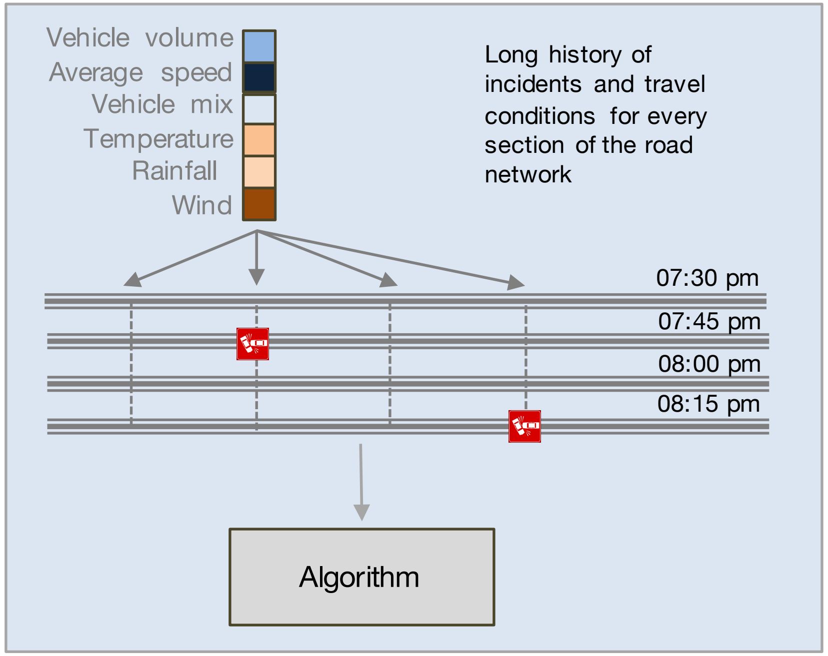
A high level view of the project logics

The project explores multiple options of algorithms to predict the status of a road section in a time future time interval (e.g. 30minutes or one hour into the future) in two categories:

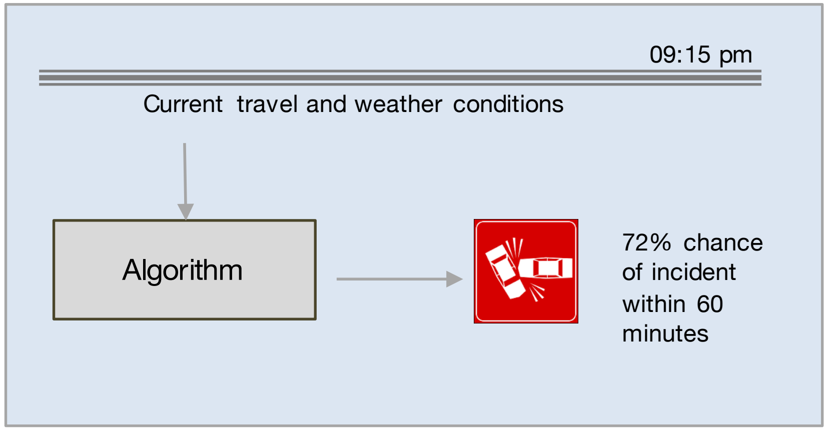
- Incident (at least one incident occurs in the future time frame)
- No incident (there isn't any incident in the future time frame)

The figure to the right illustrates in a qualitative manner the logics of the project. The data items used in the images are for illustration purposes only.

TRAINING



USING



Definitions and project hypotheses

Definitions

- **Prediction** means estimating the status probability of road section S in time interval T with respect to the occurrence of incident of type I (incident type I does not occur or occurs ones or more times).
- **Baseline** means the prediction for S-T-I based only on past incident data.

Hypothesis

- **H1:** For at least some S-T combinations, NNs can learn to reproduce the baseline incident probability for all type of incidents (not only road accidents).
- **H2:** For incidents for which there is a known qualitative relationship between data and incident, NNs are capable of predicting above baseline for multiple S-T-I combinations.
- **H3**: higher prediction accuracy is associated to a stronger qualitative relationship between influencing factors and incident.

The project provides evidence in favor and against these hypothesis.

2

INCIDENT PREDICTION WITH NEURAL NETWORKS: PREVIOUS STUDY (M0)

Summary of previous results

- Stichting Sensible Future has carried out a trial project for incident management in 2015.
- The project used a standard neural network and trained the network with one year (2010) of weather, traffic and incident data data on ±48km of Highway A12 south of Utrecht.
- The traffic/weather data is used to construct traffic "images", which are tagged with a binary incident/no incident class depending on the occurrence of an incident in an adjacent future time window. The duration of these images is 1 or 2 hours.
- 80% of the images are used for training and 20% for testing the accuracy of the network.
- The network is tested on the full road section (the test road on A12) and on segments of 1/2 , 1/4 and 1/8 of the full length.
- The study indicated that significant correlations between loop data and traffic incidents can be detected by Machine Learning algorithms and explored to predict incidents.
- For long stretches of highway (10s of km), Deep Learning Neural Networks have been shown to predict incidents with higher accuracy than average probabilities of incidents.
- This confirms that the network can detect patterns of traffic that increase the probability of an incident. However, the accuracy of prediction degrades when focusing on shorter segments of road.

Summary of previous results (cont.)

- In the study, weather data does not appear to add significantly to the prediction accuracy. This may be explained in several ways:
  - Weather does not impact incident probability for a range of incidents
  - The quality and resolution of weather used in the tests is insufficient to explain different incident probabilities
  - Weather is already captured in traffic behavior, and therefore implicitly included in traffic as a cause
- All considered, the contribution of weather to incident probability is not yet determined in this study.

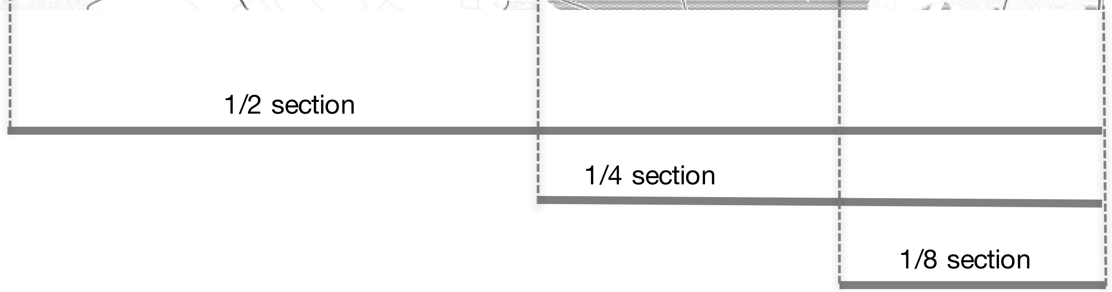
- The study also indicates a drop in prediction performance for short road segments and for short time windows. This is due to the diminishing amount of information that can be used to train the neural network, as fewer loops are included.
- To counter this information loss, it is necessary to extend the patterns in time, so that the network can use patterns in sequences of measurements to make predictions, as well as change the NN configuration, to better capture local behaviors.

Test area: highway A12/E35 next to Utrecht

The test area is a section of 48km on the A12/E35 south of Utrecht (153 loop locations).

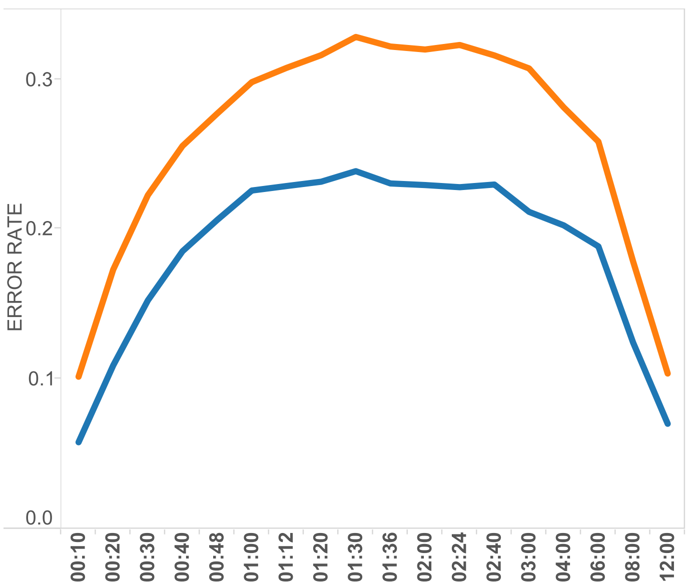
The image below shows the density of incidents on the test road in 2010 (3229 incidents in total).

- 1/2 section (88 loops, 1619 incidents)
- 1/4 section (37 loops, 942 incidents)
- 1/8 section (16 loops, 727 locations)



Benchmarking incident predictions: binary strategy

The results of the NN prediction are benchmarked against what can be achieved based solely on past incident data. The reference benchmark is called "binary strategy". This algorithm looks at the past incident data and for a time slot (e.g. 1 hour) and for a section of the road (e.g. 10km) predicts incident (no-incident) if in that time-section the historical probability of incidents was more than 50% (less than 50%). This applies to any type of incident.The algorithm does not distinguish between high or low probability of incidents, but it is the most rational strategy if only incident history is known.

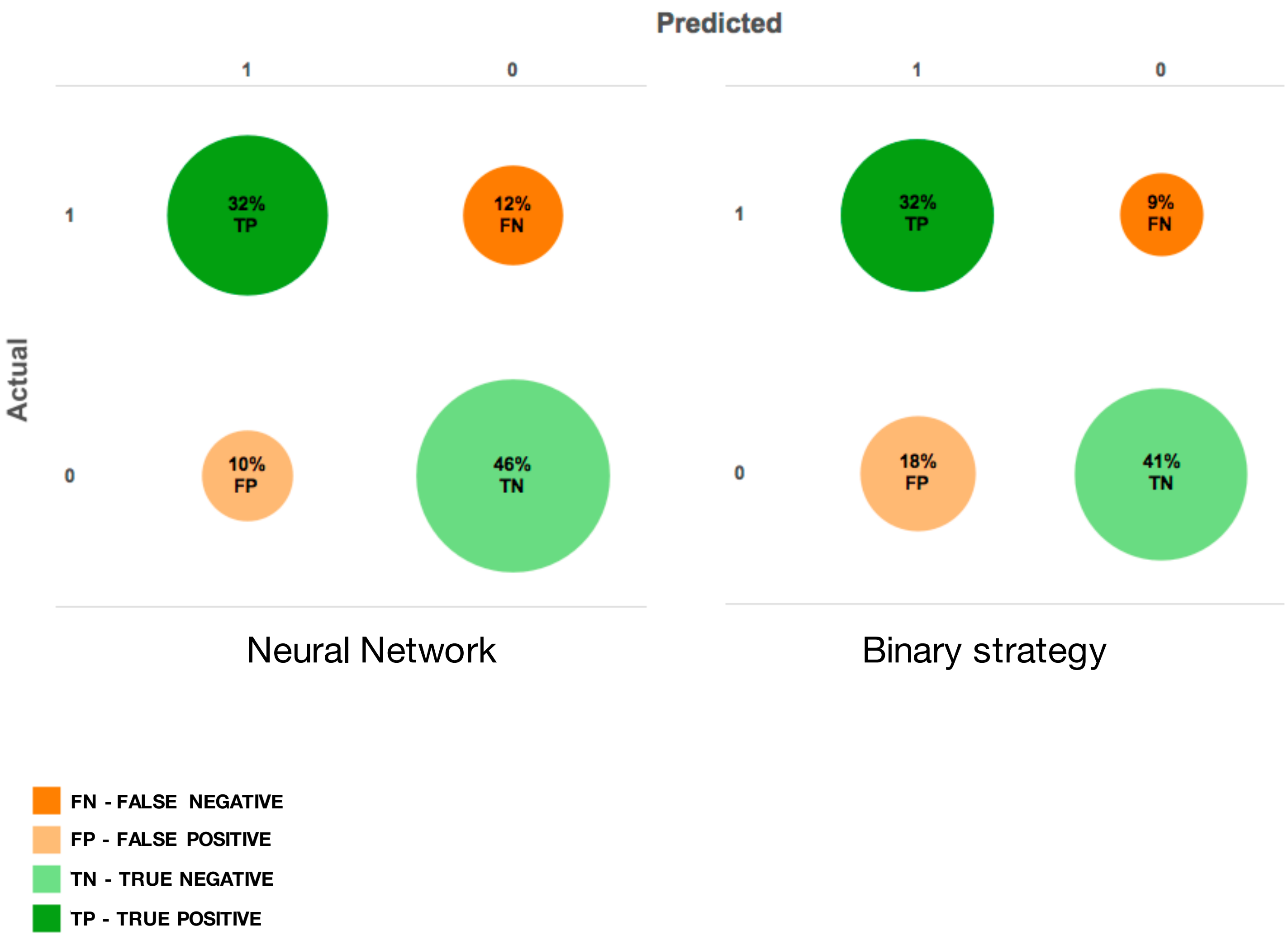


The blue line to the left shows the accuracy of the binary strategy for predicting incidents for the full test road for different time horizons:

- For short time windows the error is low (chance of incident tends to zero in short time windows)
- For long time windows the error is also low (it is almost sure that there will be an incident)
- Most errors are made in the 30mins – 6 hours time frames, which is most predictions are useful.

The orange line is an alternative strategy based on past data only (it reproduces the actual incident probability observed in the future). The binary strategy beats all other strategies based only on past data.

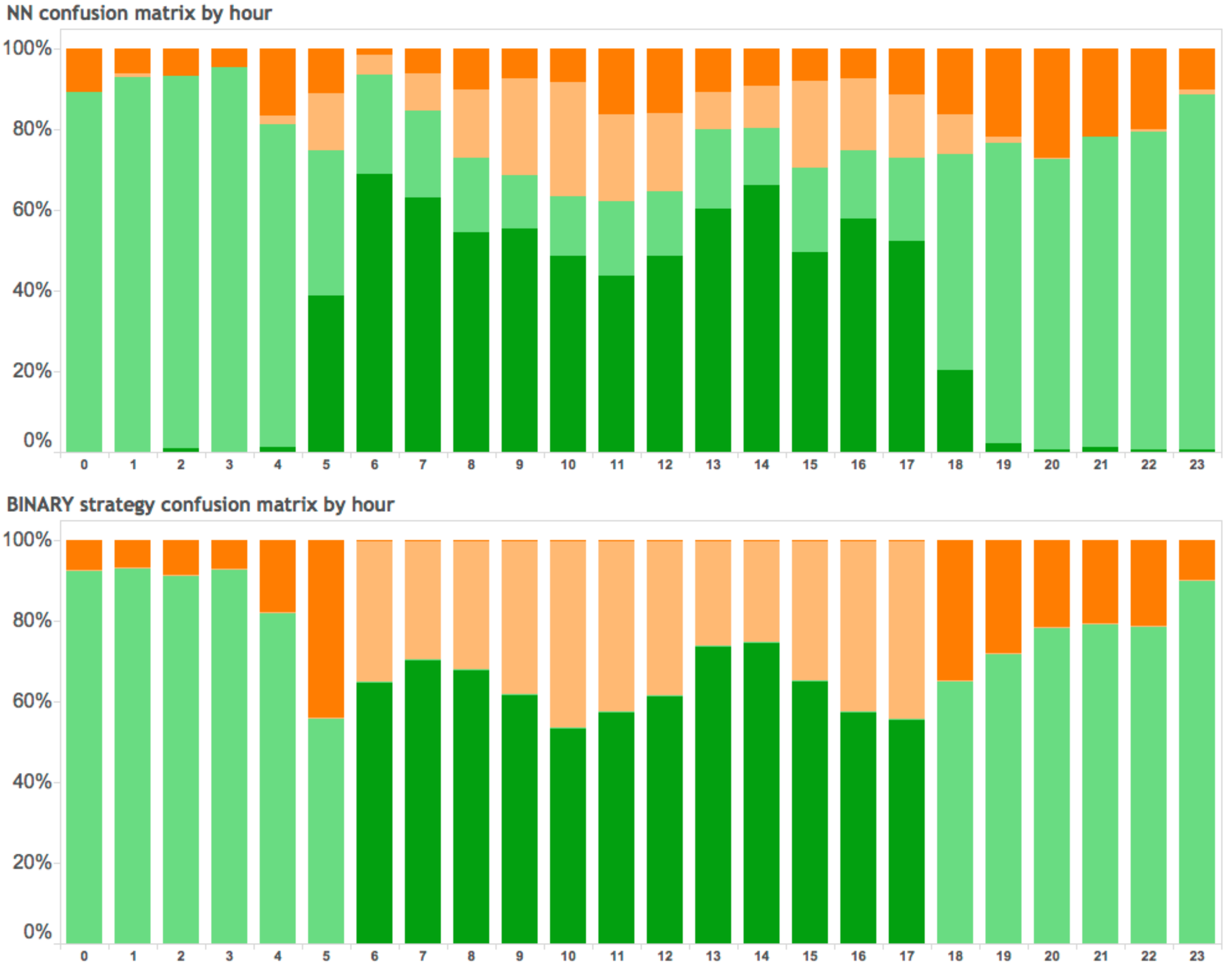
Results of previous study: confusion matrix



The diagram shows the performance of the M0 neural network against the binary strategy.

- M0 detects the same % of incidents of the binary strategy
- M0 is more precise in detecting "no incidents"
- M0 makes much less false positives (situations in which we predict incidents that did not take place)
- However, M0 makes some more false negatives (misses some incidents that did take place).

Results of previous study: confusion matrix by hour



The diagram shows the confusion matrix per hour of the day (average over 1 year).

M0 beats the Binary Strategy at all times except at 8pm and midnight, where the Binary Strategy is slightly better.

Observations and recommendations from the previous study

- Neural networks achieve a better accuracy than the binary strategy that takes only the average incident probabilities into account, indicating the existence of non-trivial traffic patterns that raise incident probability.
- The duration of the data sample appeared as too limited: data sets for 5 or 10 years would provide a much broader set of patterns against which to train the network.
- The aggregation of traffic data to 10 minutes intervals removes some potentially valuable patterns: the study advises to use 1minute data instead.
- The prediction accuracy is measured against the actual occurrences or otherwise of incidents. The neural network is trained to give the same weight to errors, be them false positives and false negatives. A different training pattern (for instance, train the network to recognize positive incident occurrences with maximum precision, accepting for instance false positives) could lead to different conclusions.
- Incident characteristics are not considered. This again is linked to the data duration which is too short to train the network on finer attributes of the incidents.

Observations and recommendations from the previous study (cont.)

1. To produce accurate localized predictions for accidents, data for one or two loops near selected incident hotspots would be most relevant. Going to the available 1min accuracy (instead of 10 min for the large scale analysis) is the best option to take full advantage of temporal patterns of the local traffic. This assumes that the data at 1m resolution is complete and coherent in time (sequences are continuous without gaps).
2. With appropriate implementation of last-generation prediction methods, the extent of the duration window and the prediction accuracy may be pushed to levels that signal applicability in real-time incident warning systems (30min-1hr, >90%).
3. Other classes of algorithms based on mining temporal patterns hold promise for local prediction for instance:
   - Convolutional networks: A class of predictive neural networks that scan data along dimensions introducing the opportunity of encoding time dependence and space correlations.
   - Recurrent Neural Networks: A class of predictive neural network with internal memory that allows it to be trained with sequences of data points and classify them in real time.
   - Bootstrapped Exponential Weights: A sequential learning family of algorithms that combine the predictions of a large number of data-derived predictors into a single decision of higher accuracy.

The study suggests to test these algorithms with 5 years of traffic and incident data at 1min resolution, with the goal of raising the accuracy to levels that open the prospect of application to real-time streams of loop data.

3

DESIGN OPTIONS

Setting up the prediction strategies

The number of implementation options is very large since a variety of neural network designs can be used, and within each design, a large array of configurations (hyper-parameters) can be tested.

For practical reasons we looked at three strategies:

- Strategy 0 (the strategy used in M0). The neural Network (NN) learns from all inputs simultaneously. This is called a fully-connected network.
- Strategy 1: the NN first tries to extract features from data (learns the patterns) and then figures how to associated features to incidents. This is similar to the way we interpret images and has successfully been implemented in image, speech and handwriting recognition, among others.
- Strategy 2: the NN learns local conditions that lead to incidents by capturing explicitly the time evolution of traffic conditions. This is similar to the way we understand speech and has been successfully implemented in time series prediction.

These strategies are reflected in the prediction canvass (see below) which maps strategies and different type of predictions.

Strategies: overview

Strategy 0
Learn from all inputs simultaneously

Fully Connected Network

M0

- The reference, fully connected, network
- Requires massive amounts of data to train well
- Large networks become very complex and extremely time and resource consuming.

Strategy 1
First learn data features, then incidents

Convolutional Network

M0+

- Use the same data of M0 but a different learning strategy
- Less computationally intensive, learns faster

Convolutional Network

M1

- Use the same model of M0+
- Higher resolution data
- Longer data history

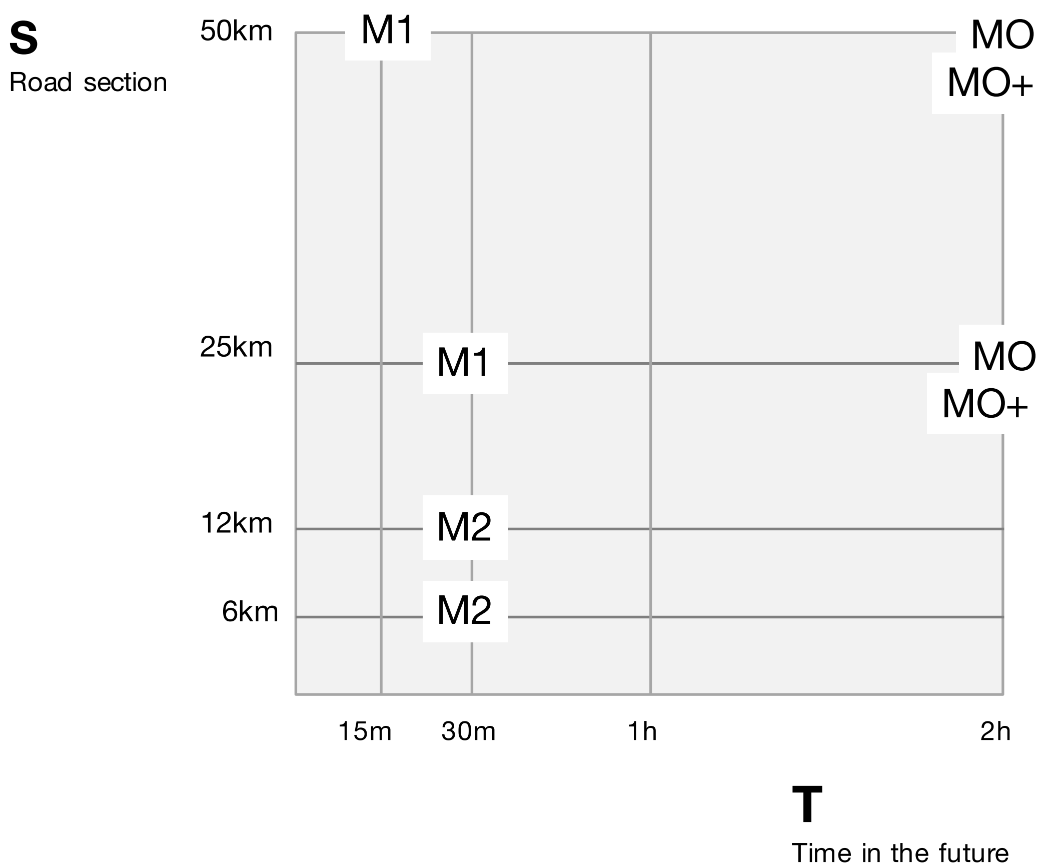
Strategy 2
Learn local conditions that lead to incidents

Recurrent Neural Network (LSTM)

M2

- Uses the same data of M1
- Shorter road sections
- All local features are included

Prediction canvas



The project looks at selected combinations of S (length of the road section) and T (time window for prediction).

The two dimensions are:

- T = Time window in the future. The range is between 0-15 minutes to 0-2 hours in the future.
- Road section length, between 6km and 50km.

The networks tested are indicated with M0, M0+, M1, M2.

Design parameters for the networks

	M0	M0+	M1	M2
	NN applied on A12 section	Convolutional NN with same data setup of previous study (M0)	Convolutional NN with new data - A12 section (1 min)	LSTM NN for hot spots on - A12 (1 min)
Input data	Traffic and weather data	Traffic data	Traffic data	Traffic data
Traffic features used	Speed, Flow, Temperature, Wind, Rainfall	Speed, Flow	Speed, Flow	Speed, Flow
Incident features used	Incident, no incident	Incident, no incident	Incident, no incident Specification of incident type	Incident, no incident
Length of the test road section	48km, 24km, 12km, 6km	48km, 24km	48km, 24km	6-10km
Future interval	1hour, 2 hours	1 hour, 2 hours	15m, 30m, 1hour, 2 hour	15-30 min
Training data	1 year	1 year	4 years (excluding gaps)	4 years (excluding gaps)
Loop data processing	Average across directional lanes	Average across directional lanes	Average across directional lanes	Original measurements
Data granularity	10 minute averages, rolling	10 minute averages, rolling	1 minute data	1 minute data
Type and design of the NN	Fully connected NN	Convolutional	Convolutional	LSTM
Input layer design (e.g. 1hour rolling 6*10min sections)	1 hour data, rolling six 10 -minute sections	1 hour data, rolling six 10-minute sections	1 hour data, rolling 1-minute	1 hour data, rolling 1-minute
Output type	Binary (incident, no incident)	Binary (incident, no incident)	Binary (incident, no incident)	Probability of incident

M0: DATA MODEL

The data is structured into one-hour traffic sets, labelled with the incident-no incident occurrence *in the next hour* (the time window T).

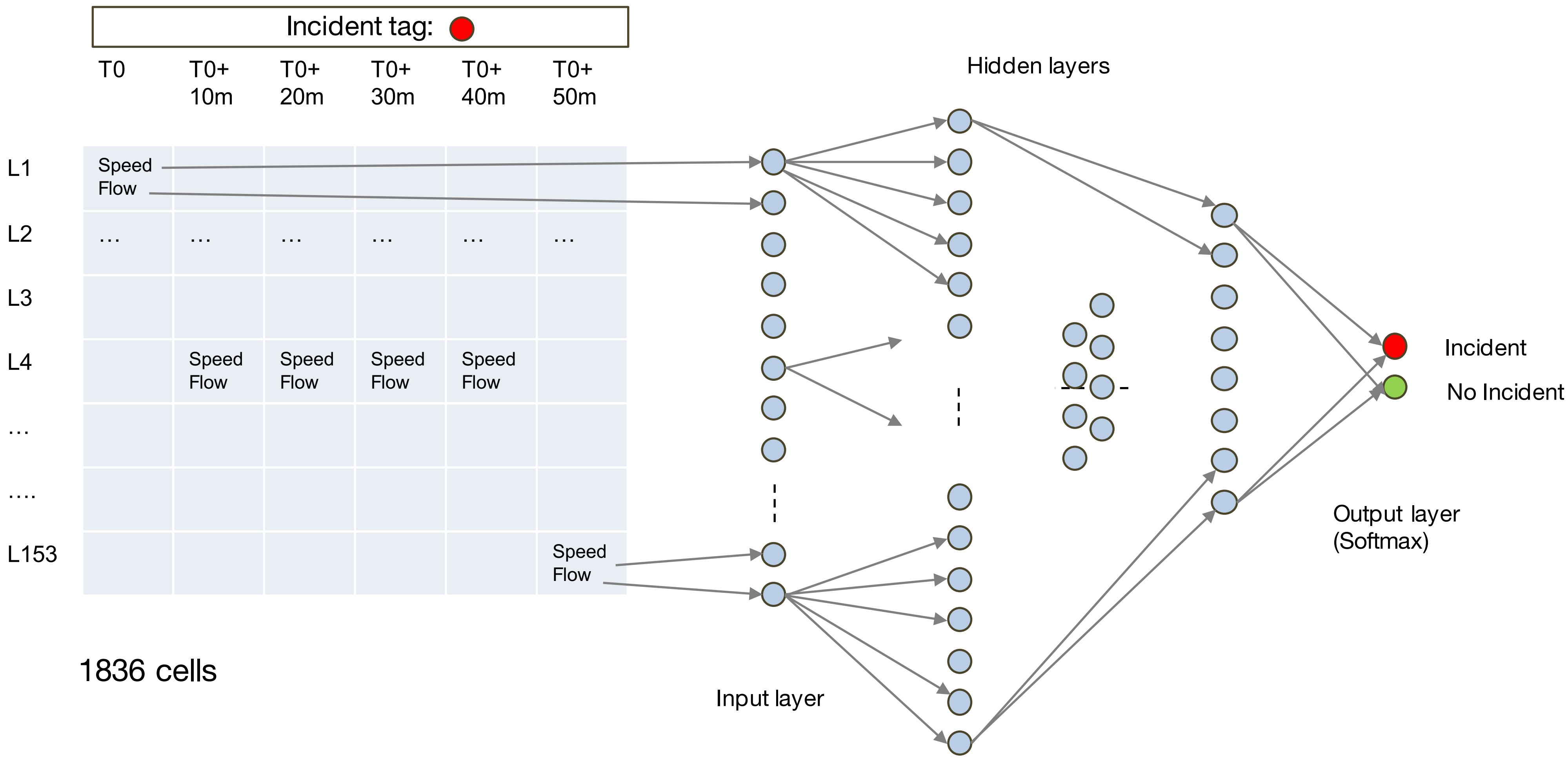
For 1 hour training "image", we have:
- 153 loop locations
- 2 variables per loop location
  - Average speed across the lanes
  - Average flow across the lanes
  - Average over 10 minutes
  - 6 time slots of 10 minutes

1836 input data points for one hour: they are fed to the input layer of the NN

In one year of data:
- 1 image every 10 minutes (sliding image): in total 52,560 training images
- Each image is tagged with "incident", "no incident" if in the next hour there is an incident/no incident
- 52,560 is the maximum theoretical number of images. We excluded images that had data gaps (missing loop data). Remaining images are 36,601.
- One incident can tag multiple images because of the sliding image setup.

- The same setup can be generalized for 2 hour images or for shorter, 30 minute images.

M0: neural network simplified diagram



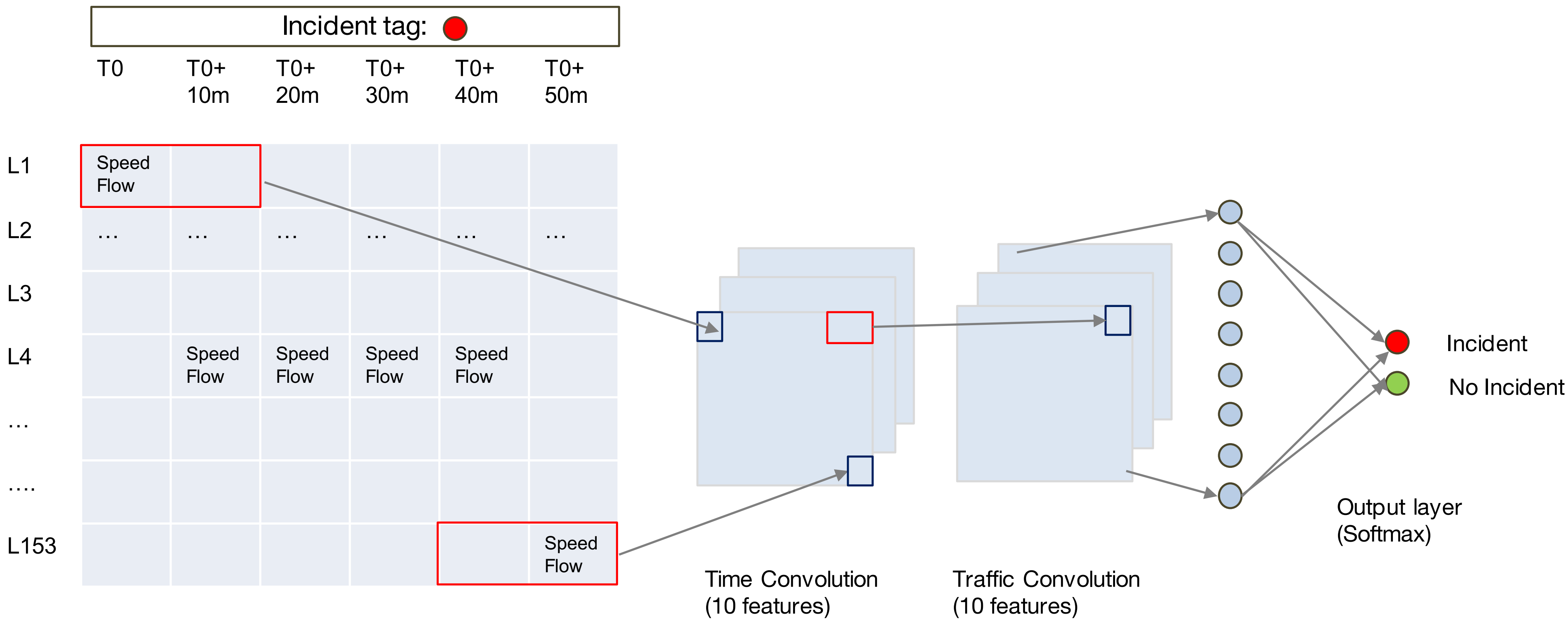
M0+: DATA MODEL

MO+ uses exactly the same data used in M0, the difference being that the neural network that uses the data has a different structure: a convolutional network.

Use two sequential convolution layers:

- The first is a time convolution for speed-flow (kernel of two time slots).
    - In this layer two consecutive data points are proceed by a convolutional formula.
    - The formula captures different variations of traffic (speed and flow separately)
    - From this convolution we derive 10 features.
- The second is a traffic convolution for speed-flow (kernel of two neurons)
    - It applies to the features derived from the first convolutional layer
    - It is inspired by level of service (the feature extracts combinations of speed-flow and creates a speed-flow space)
    - From this convolution we derive 10 features

M0+: neural network simplified diagram



1836 cells

M1: DATA MODEL

M1 is a variation of M0+ where we use shorter training images to predict incidents in the next 30 minutes. This model is applied to the new dataset:

- 172 gates (groups of loops across multiple lanes)
- 2 variables per gate (average speed and average flow across lanes)
- Minute data, leading to 30 time slots per training image of 30 minutes

The input image now contains 10,320 cells.

Use three sequential convolution layers:

- Time convolution for speed-flow (kernel of two time slots)
    - Captures the variation of traffic (speed and flow separately)
    - 5 features
- Traffic convolution for speed-flow (kernel of two neurons)
    - Inspired by level of service (the feature extracts combinations of speed-flow, creates a speed-flow space)
    - 5 features
- Space convolution for loops (kernel of 2 successive loops)
    - Inspired by dynamics of traffic in space
    - 5 features

M2: DATA MODEL

M2 belongs to a different class of Neural Networks, called Recurrent Neural Networks (in this LSTM). The network design is as follows:

- 10 loops per traffic direction
- 2 variables per loop
- All lanes are considered
- Minute data, leading to 30 time slots per training image of 30 minutes

The strategy is to feed the images to the the network. However, major gaps in the data make this strategy unfeasible based on raw data. The reason is that broken sequences (sequences for which there is a missing data item) are interpreted by the network and the missing data is included in the predictions, which it's only noise.

Several tests with the existing data have led to results that are not superior to the Binary Strategy, indicating that M2 is learning the averages but fails to detect patterns that are early indicators of incidents.

The explanation is the data gaps, which confuse the network. To apply NNs to local predictions the input data first needs to be complete or imputation needs to be applied to the data.

4

DATA PREPARATION AND SYSTEM SETUP

The traffic data set

The loop datasets, provided by NDW, contains three types of data relevant to the study:

1. Loop measurements per 1 minute. We use complete records for the years 2010-2014.
    - Loop id: the unique identifier of the loop in the road network
    - Index of lane number: the lane numbering, starting with # for the left-most lane
    - Average vehicle speed (km/h)
    - Car flow (# of vehicles in the last min)
2. Lane index lookup table
    - Indicates the distinction between lanes for loops
3. Loop metadata
    - Loop locations plus other loop metadata

Data is available for two area. The tests have been carried out on A12.

A12 section south of Utrecht

Data completeness: number of traffic measurements in time

The diagram shows the number of "good" reads for traffic loop data per day in the period 01/2010 – 12/2014

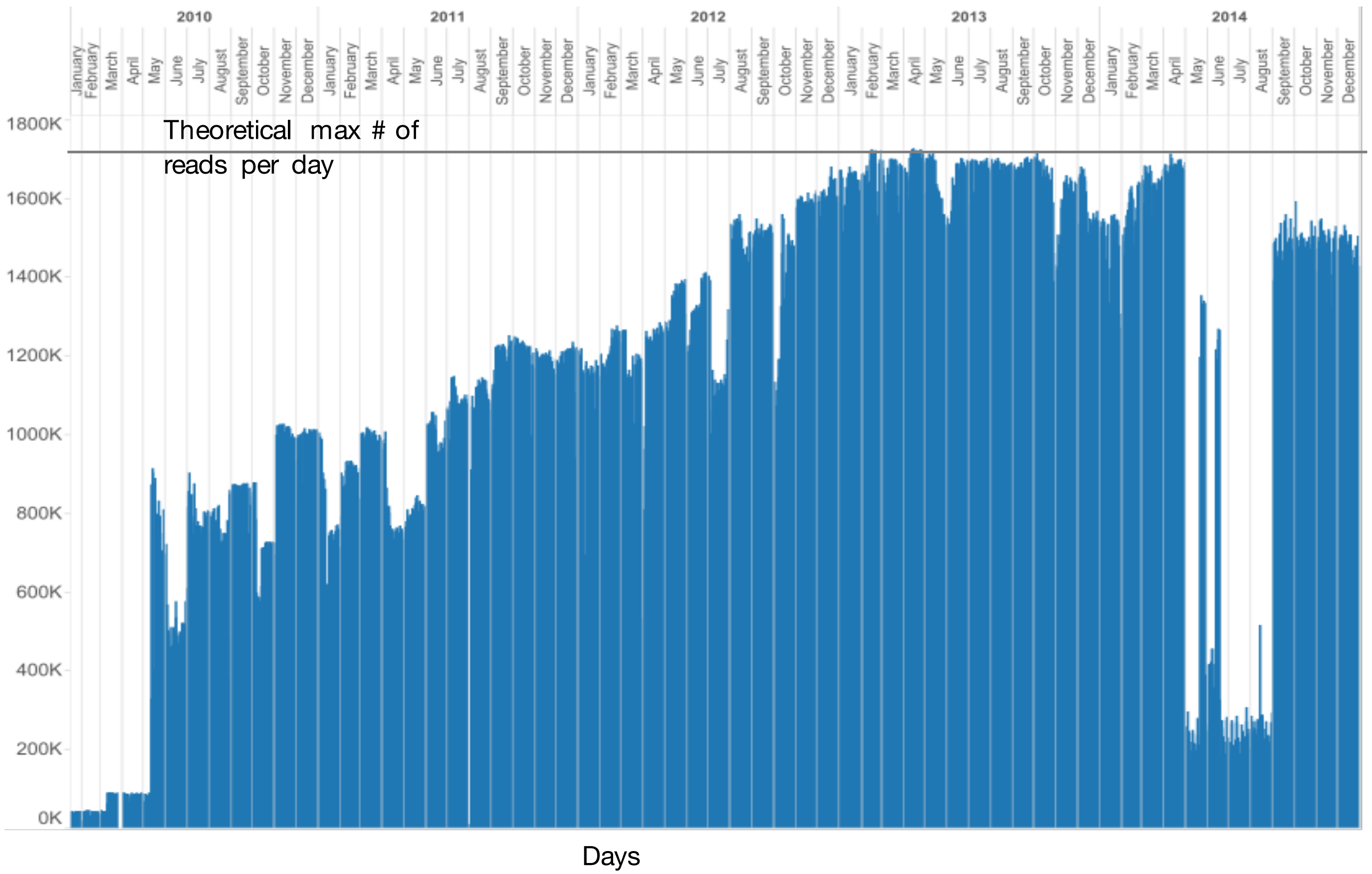
The original data format is

- speed-flow
- per loop-lane
- per minute

On average, in this dataset, assuming all loops generate good data continuously, we would have 1,715,040 maximum reads per day.

The average of good reads over 5 years in this dataset is 1,140,465.

The % good reads per day is 66.5%.



EDA for new dataset

The following steps were carried out to prepare the data for the network training.

- Step 1 - Data selection traffic dataset
  - From the original CSV file we select flow and speed for the category “any vehicle”: we do not make any distinction between vehicle types
  - For the test on A12 we focus on the main carriageway
- Step 2 - Data pre-processing for traffic
  - On A12: original # of loops is 551 (both directions, test area)
  - If tow loops have different ID but same location, they are merged: the result is 349 loop locations (at times they are called *gates*)
  - Speed and flow are averaged across lanes: each location takes one speed and one flow measurement
- Step 3 - Data preparation for incident data
  - Select incidents within the road section
  - We use date-time of the incident only
  - We use three broad incident classes: vehicle problems, accidents and other.

5

TESTS AND RESULTS

MAIN RESULTS: NOTES ON INTERPRETING THE RESULTS

We have carried out multiple tests on various combinations of S (road length), T (time window for prediction) and I (Type of Incident) for incident data aggregated at 10 minutes and 1 minute.

The table in the next page shows the main results.

For each test we show the results of the NN and the results of the corresponding Binary Strategy.

In cases when the NN performs equal (or worse) compared to the benchmark, we have indicated the cell with an “x” indicating a non viable test result.

If a specific combination of S-T and I has not been tested, the table shows a “-” mark.

All results refer to the test area on the A12.

The series of images that follow the table illustrate the confusion matrix for the various strategies. The confusion matrix shows four classes:

- TP - True positives (incident was predicted and it did occur)
- TN - True negatives (incidents were not predicted and incidents did not occur)
- FP – False positives (incident was predicted but did not occur)
- FN – False Negative (incidents was not predicted but an incident occurred)

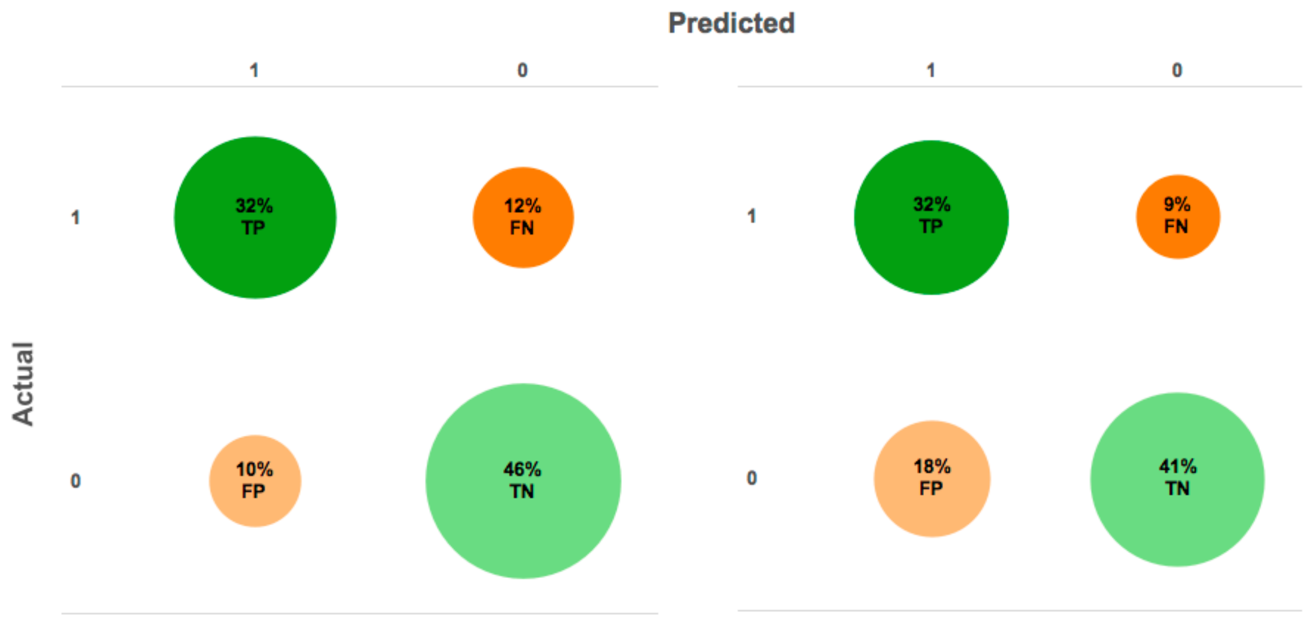
CNV stands for Convolutional Networks.

M2 is not included (see previous notes on M2).

MAIN RESULTS

		10m traffic data			1m traffic data	
S Test Segment on A12	**T** Prediction time window	**M0** Original Study	**MO+** CNV on original study data	**Binary predictor benchmark**	**M1** CNV 1m data	**Binary predictor** benckmark
1 (full length)	2h	78%	83%	76%	71%	65%
1 (full length)	30m	-	-	-	69%	62%
1 (full length)	15m	-	-	-	73% (all incidents)	69%
1 (full length)	15m	-	-	-	76% (accidents + vehicle problems	69%
1 (full length)	15m	-	-	-	81% (accidents)	65%
1/2	2h	76%	80%	79%	66%	62%
1/2	30m	-	-	-	73%	68%
1/4	30m	-	-	-	-	-
1/4	15m	X	-	-	X	-
1/8	15m	X	-	-	X	-

M0 (previous study)



Neural Network

Binary strategy

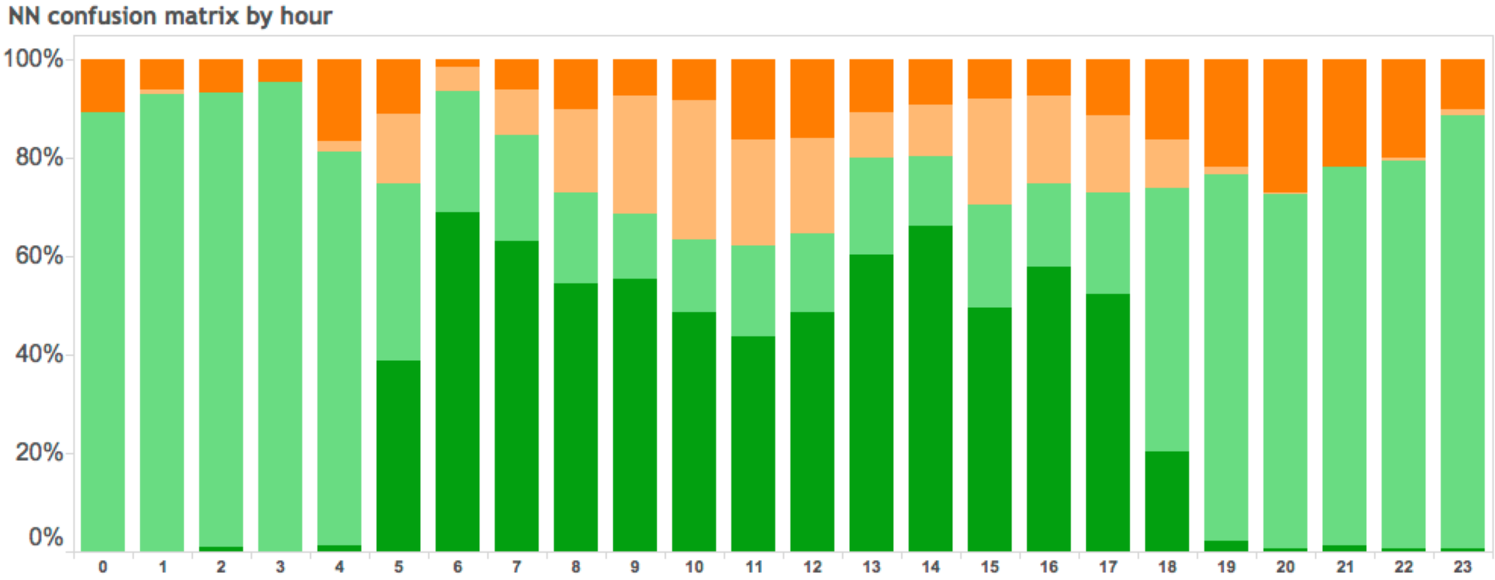
- FN - FALSE NEGATIVE
- FP - FALSE POSITIVE
- TN - TRUE NEGATIVE
- TP - TRUE POSITIVE

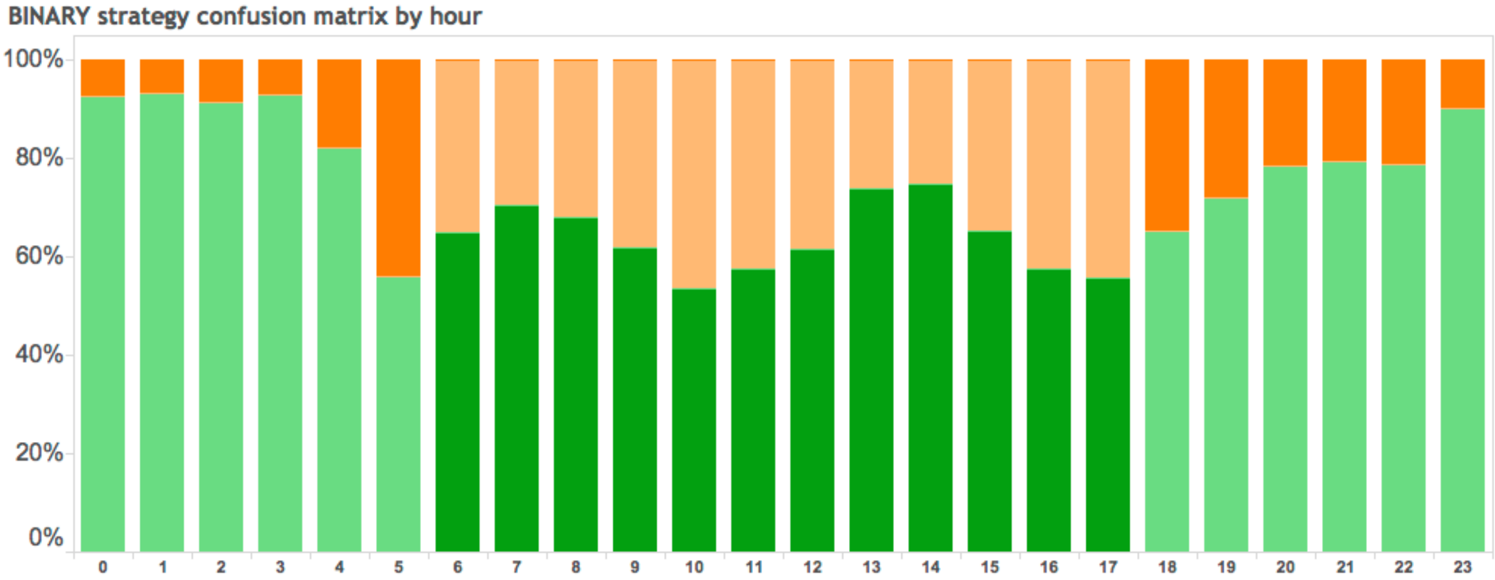
S	Test area: 48km
T	2 hours
I	All incidents

The diagram shows the performance of the M0 neural network against the binary strategy.

- M0 detects the same % of incidents of the binary strategy
- M0 is more precise in detecting “no incidents”
- M0 makes much less false positives
- However, M0 makes some more false negatives (misses some incidents that did take place).

M0 (previous study)





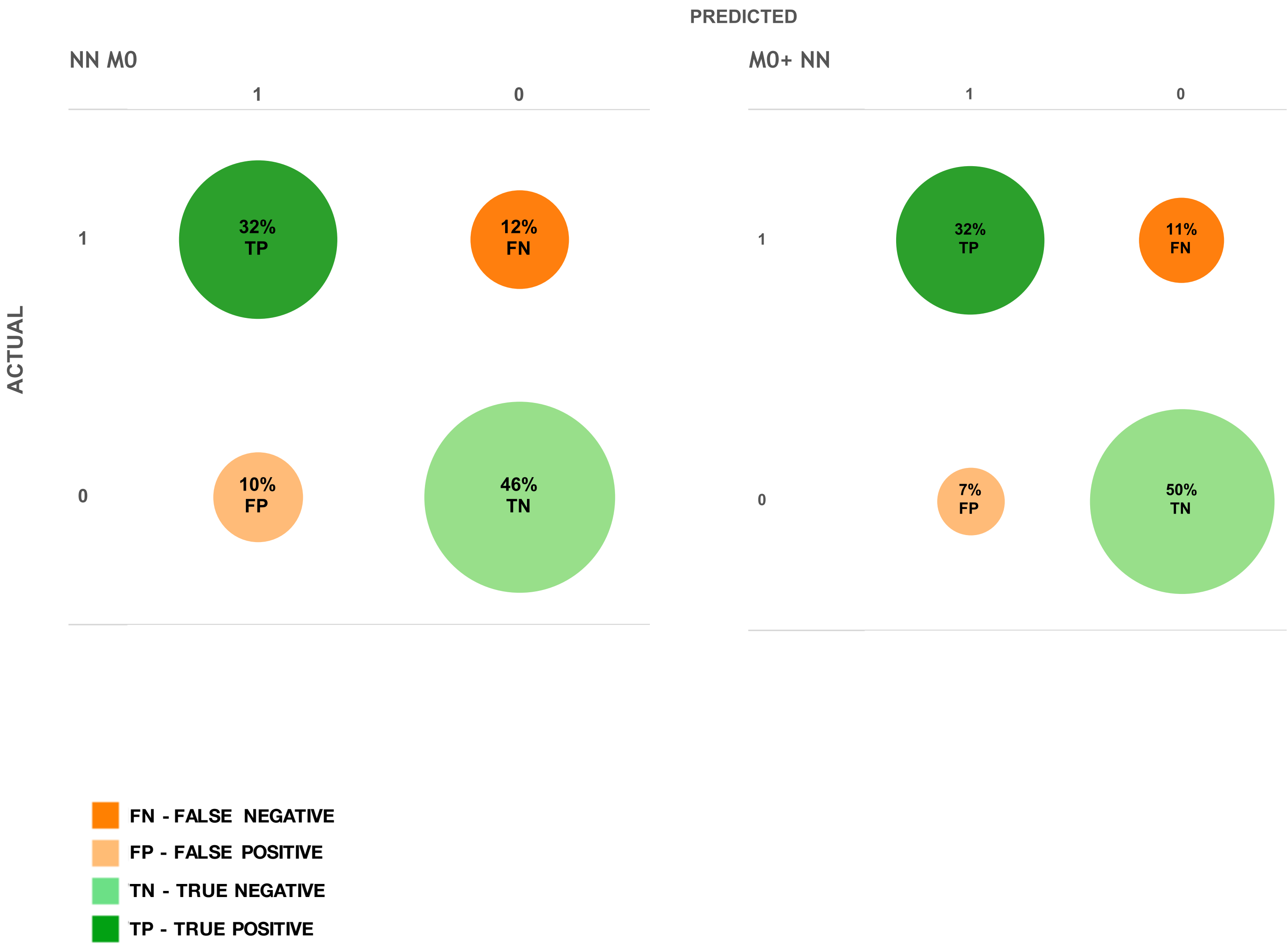
S	Test area: 48km
T	2 hours
I	All incidents

The diagram shows the confusion matrix per hour of the day (average over 1 year).

M0 beats the Binary Strategy at all times except at 8pm and midnight, where the Binary Strategy is slightly better.

FN - FALSE NEGATIVE
FP - FALSE POSITIVE
TN - TRUE NEGATIVE
TP - TRUE POSITIVE

M0 vs. M0+



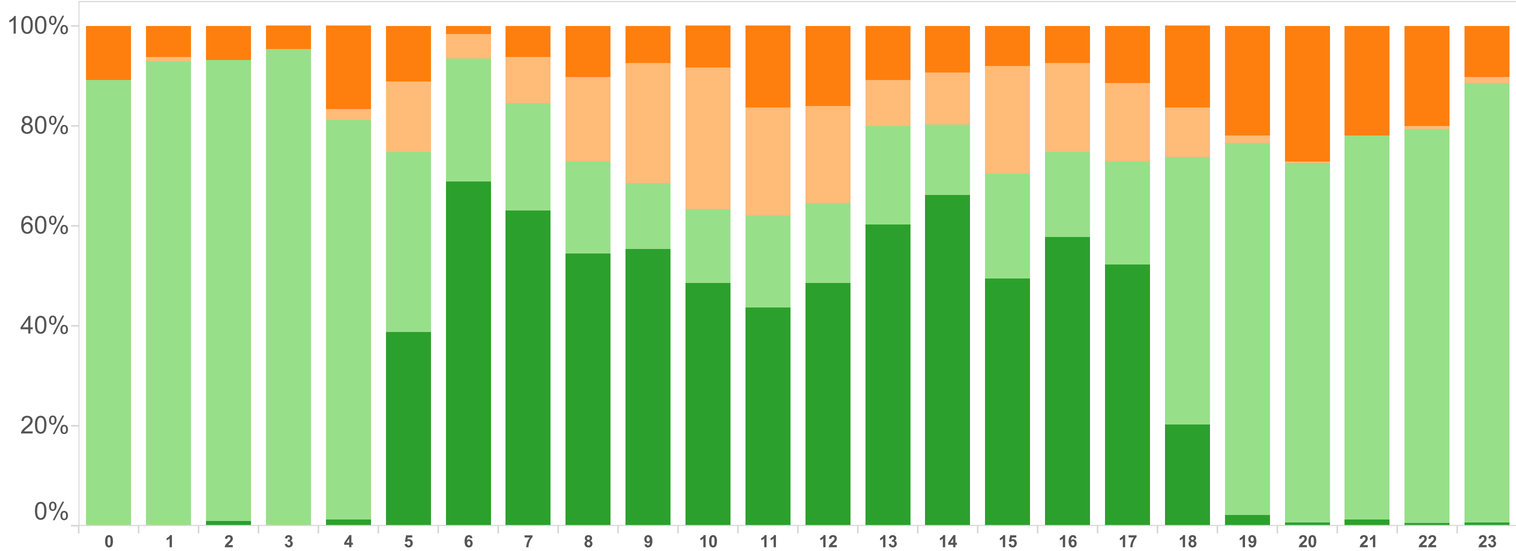
S	Test area: 48km
T	2 hours
I	All incidents

The diagram shows the confusion matrix for M0 and M0+.

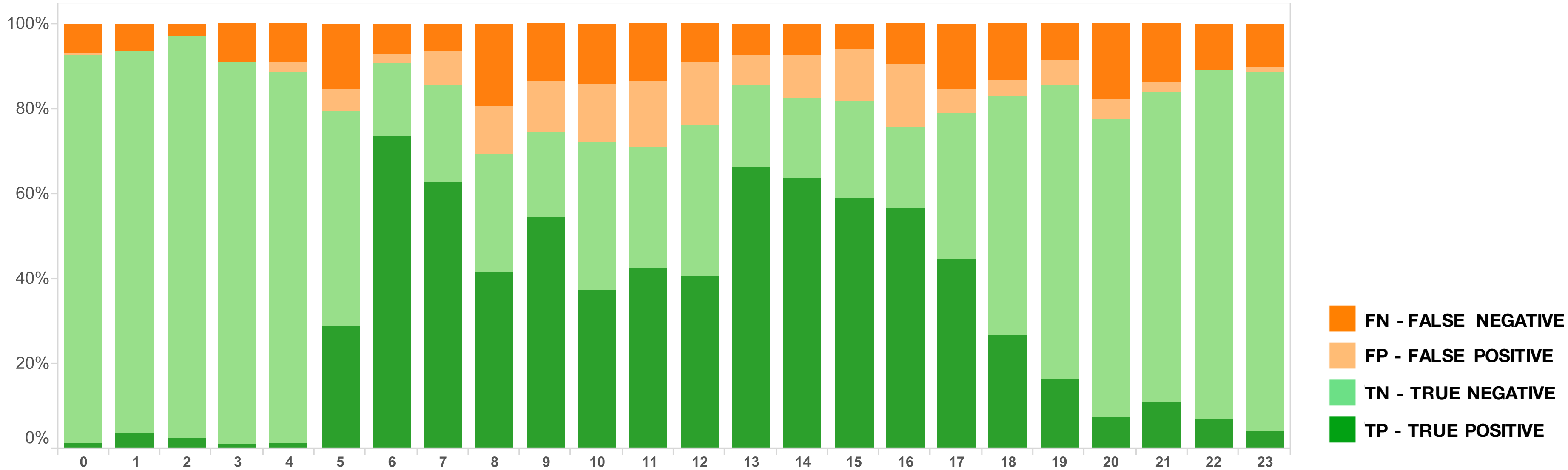
M0+ detects better true negatives and reduces false positives. The false negatives are comparable.

M0 vs. M0+

NN M0 confusion matrix by hour



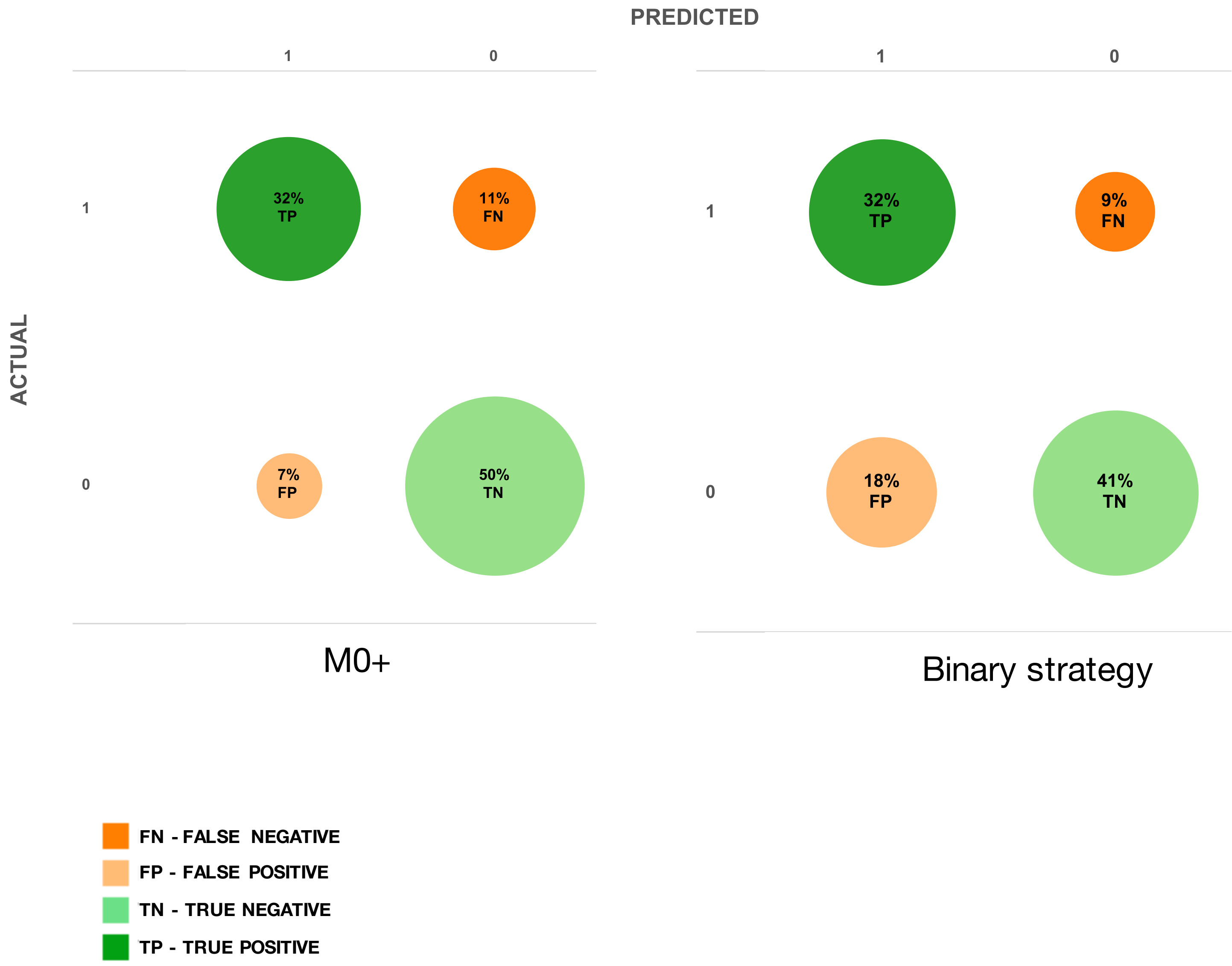
NN M0+ confusion matrix by hour



S	Test area: 48km
T	2 hours
I	All incidents

M0+ is more sensitive during night hours and balances predictions better during the day.

M0+ vs. Binary Strategy

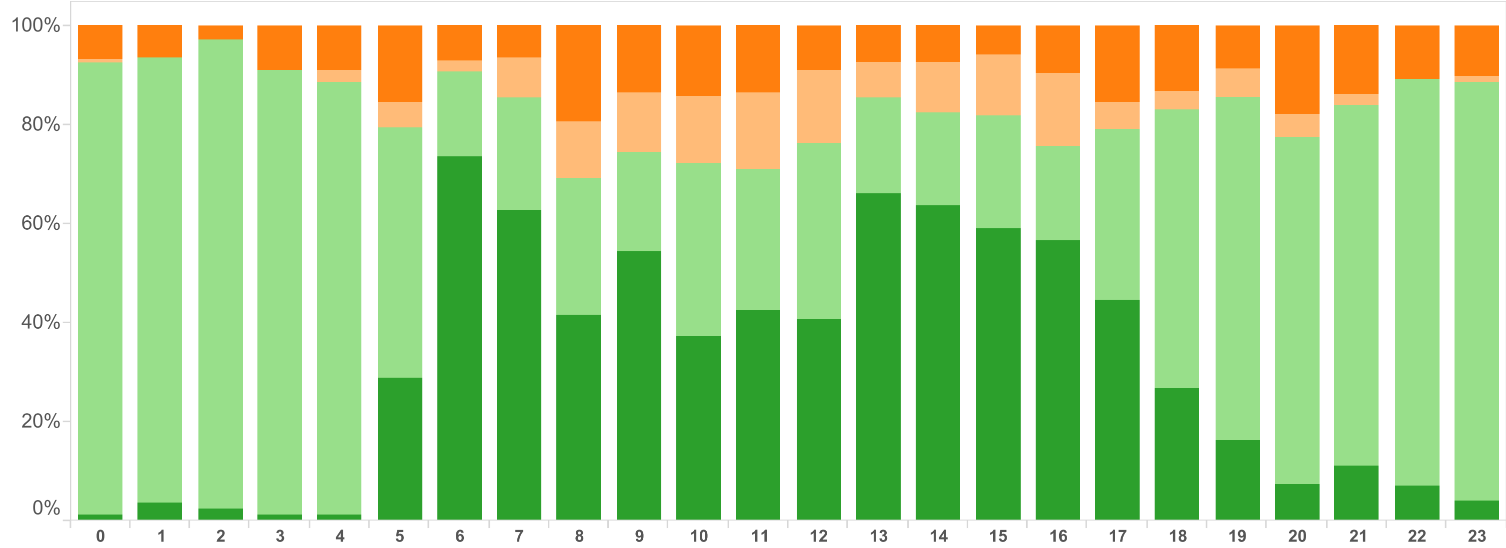


S	Test area: 48km
T	2 hours
I	All incidents

Compared to the binary strategy, M0+ scores better true negatives and significantly reduces false positives.

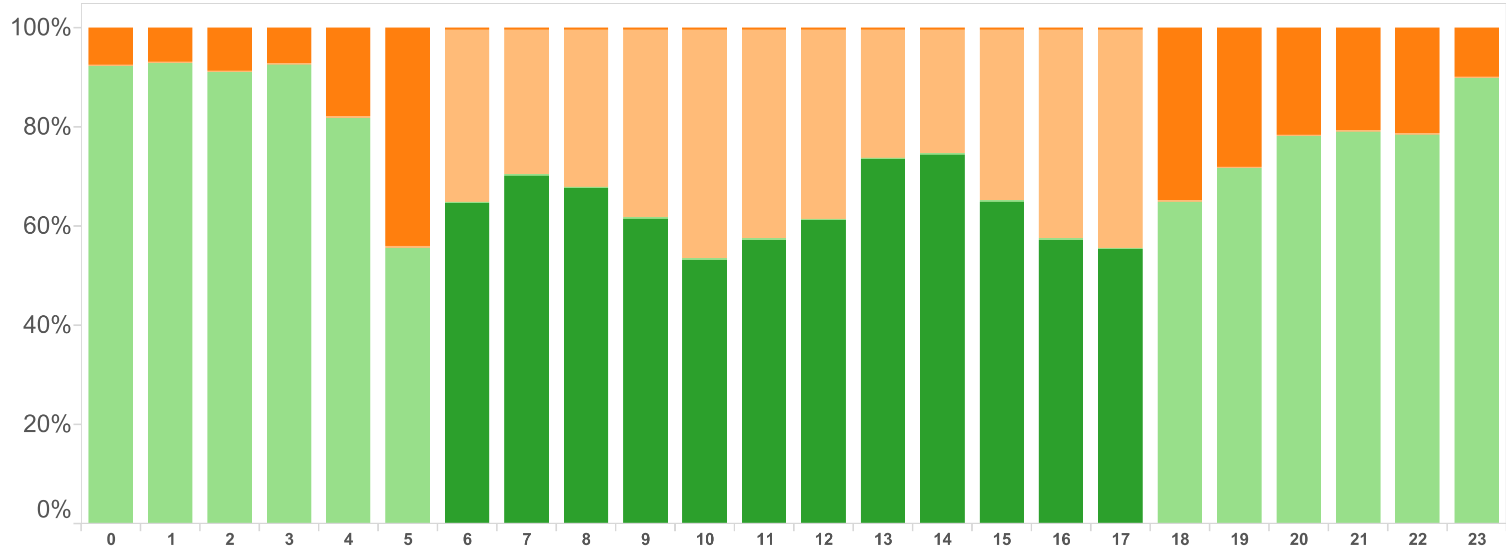
M0+ vs. Binary strategy

NN M0+ matrix by hour



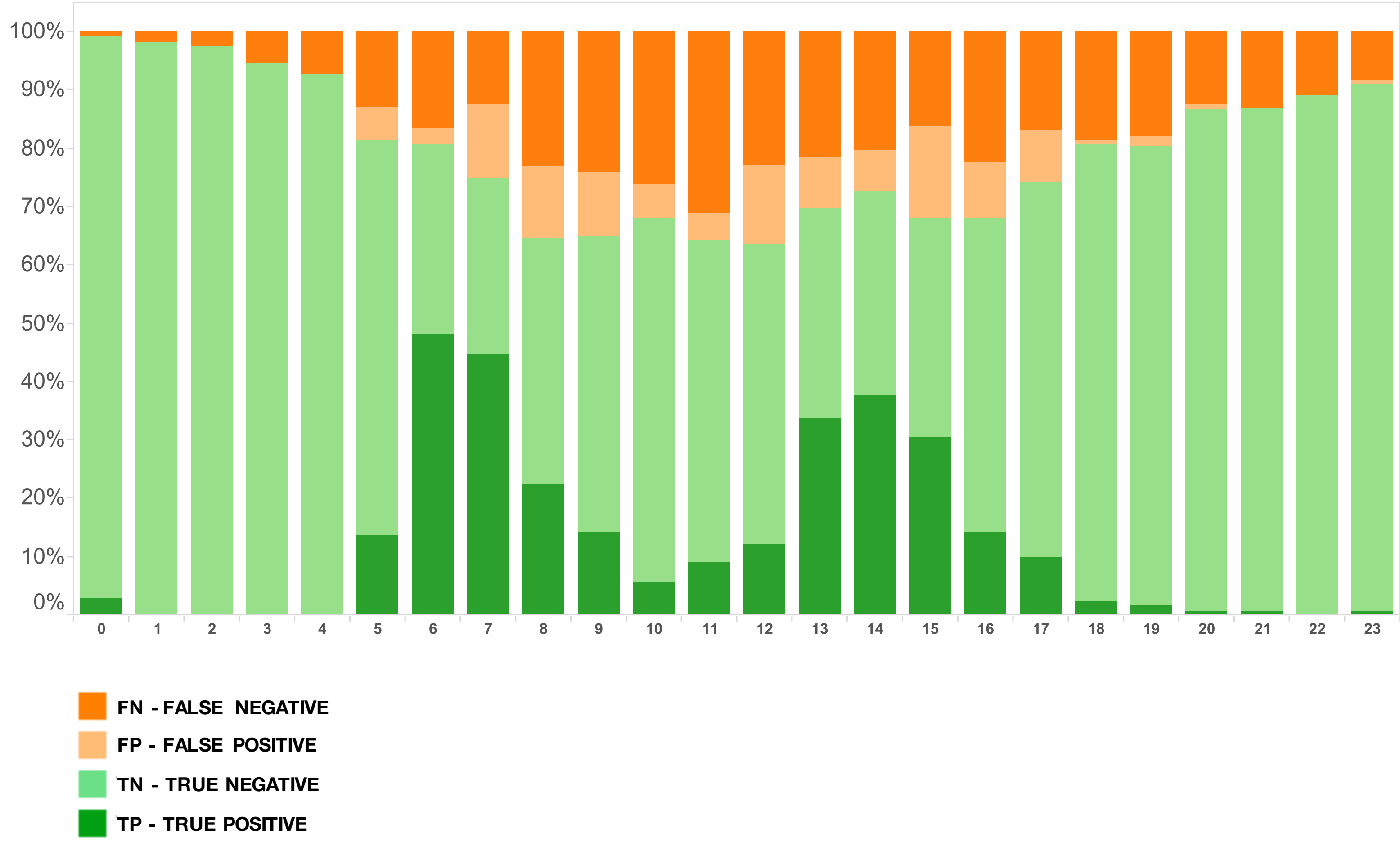
S	Test area: 48km
T	2 hours
I	All incidents

BINARY strategy confusion matrix by hour



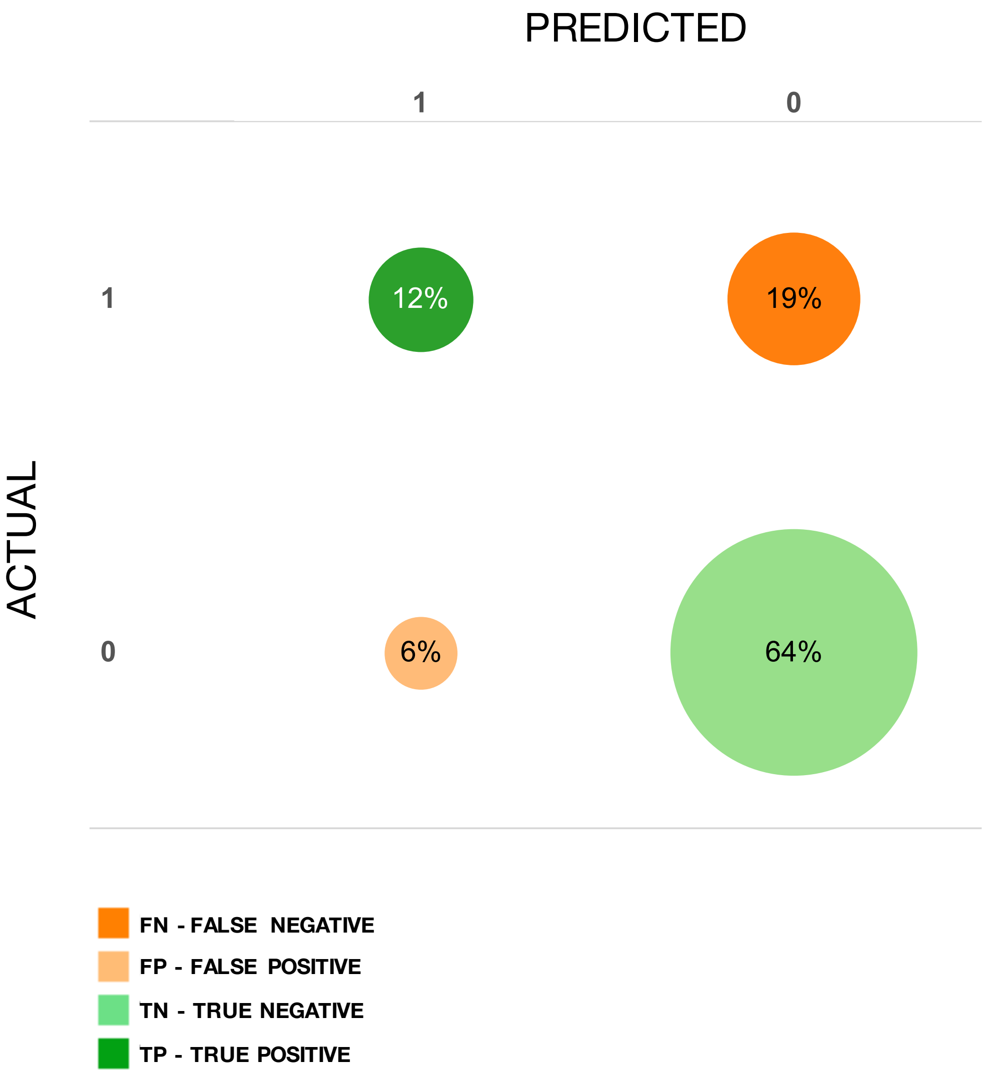
FN - FALSE NEGATIVE
FP - FALSE POSITIVE
TN - TRUE NEGATIVE
TP - TRUE POSITIVE

M0+ on reduced road section



S	24 km
T	2 hours
I	All incidents

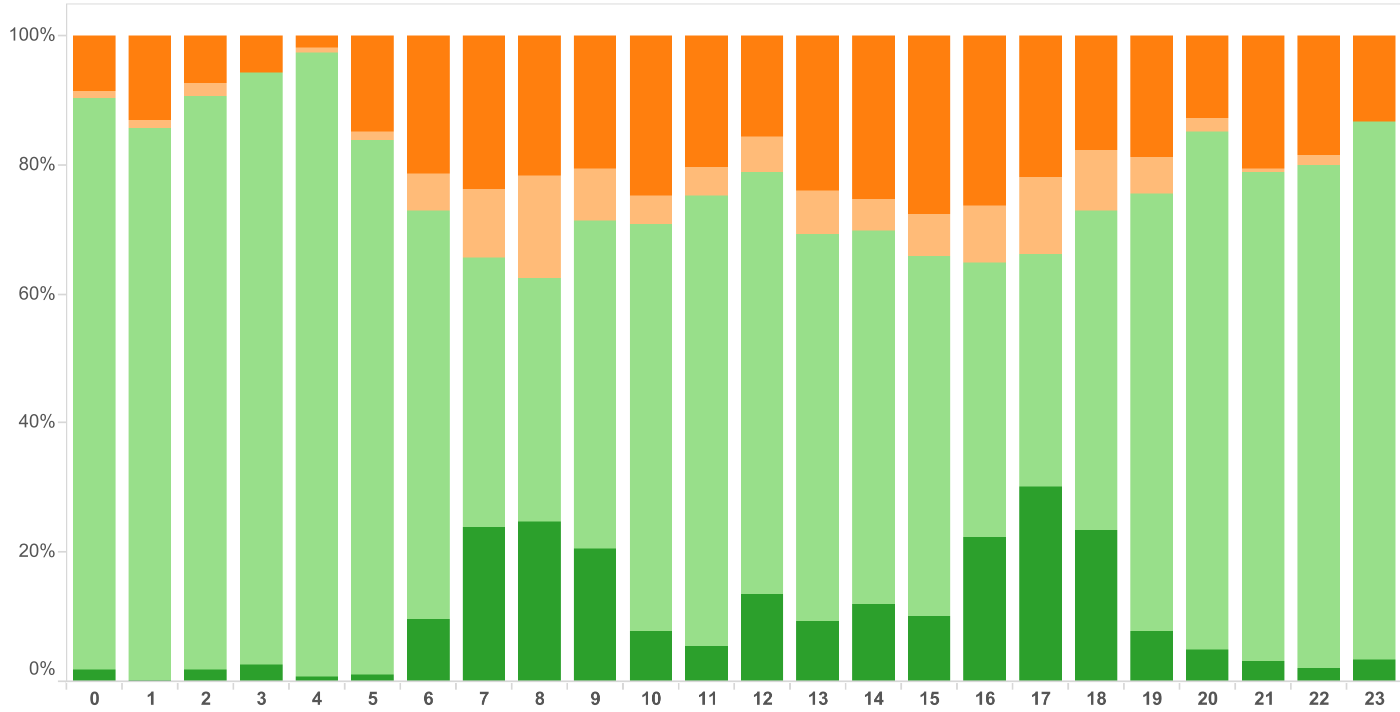
M1 on full road, 15 minutes - accidents and vehicle problems



S	Test area: 48km
T	15 minutes
I	Accidents and vehicle problems

With short time intervals, the prevailing status will be ”no incident”. M1 predicts well true negatives and makes few false positives. It does make mistakes on false negatives.

M1 on full road, 15 minutes - accidents and vehicle problems

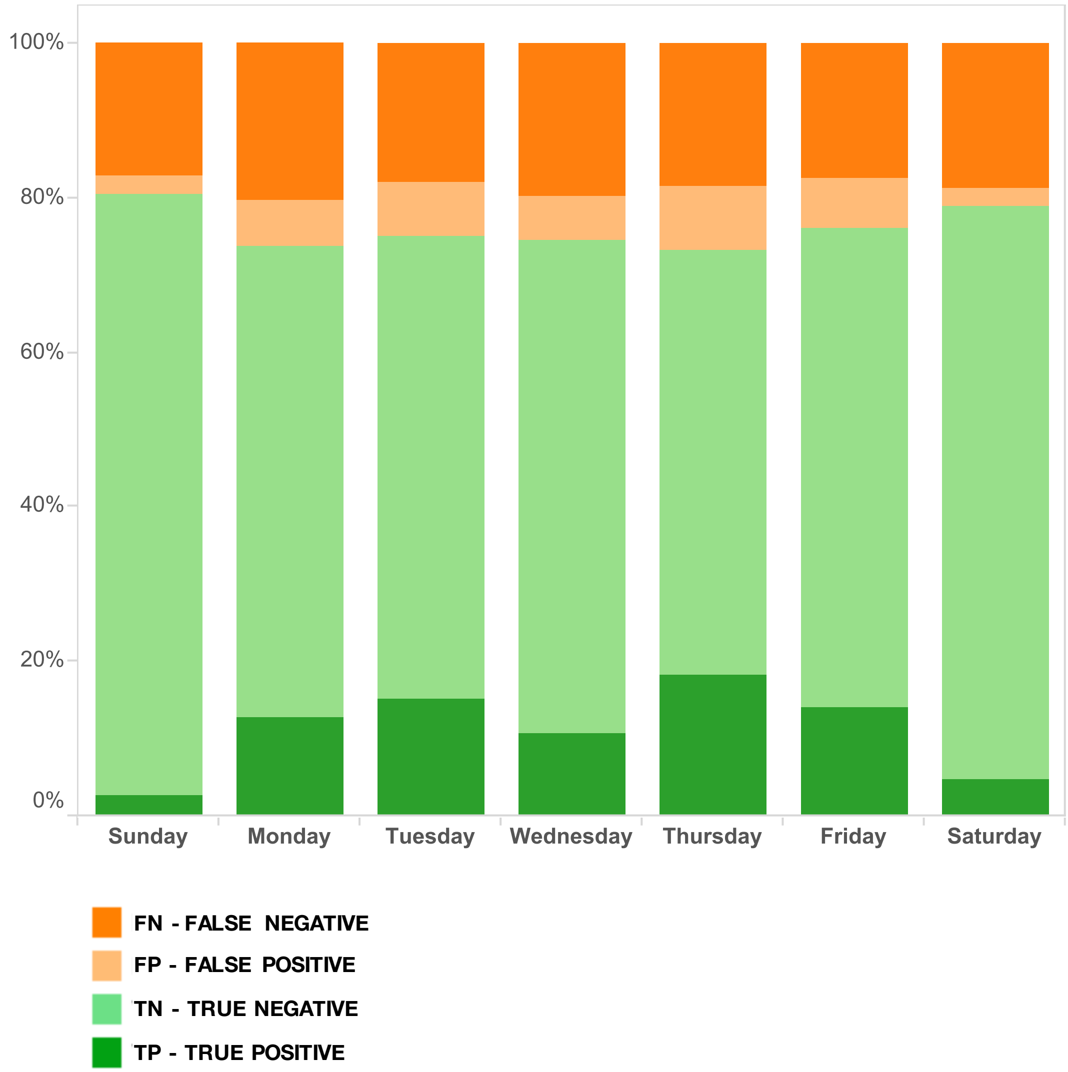


FN - FALSE NEGATIVE
FP - FALSE POSITIVE
TN - TRUE NEGATIVE
TP - TRUE POSITIVE

S	Test area: 48km
T	15 minutes
I	Accidents and vehicle problems

With short time intervals, the prevailing status will be ”no incident”. M1 predicts well true negatives and makes few false positives. It does make mistakes on false negatives.

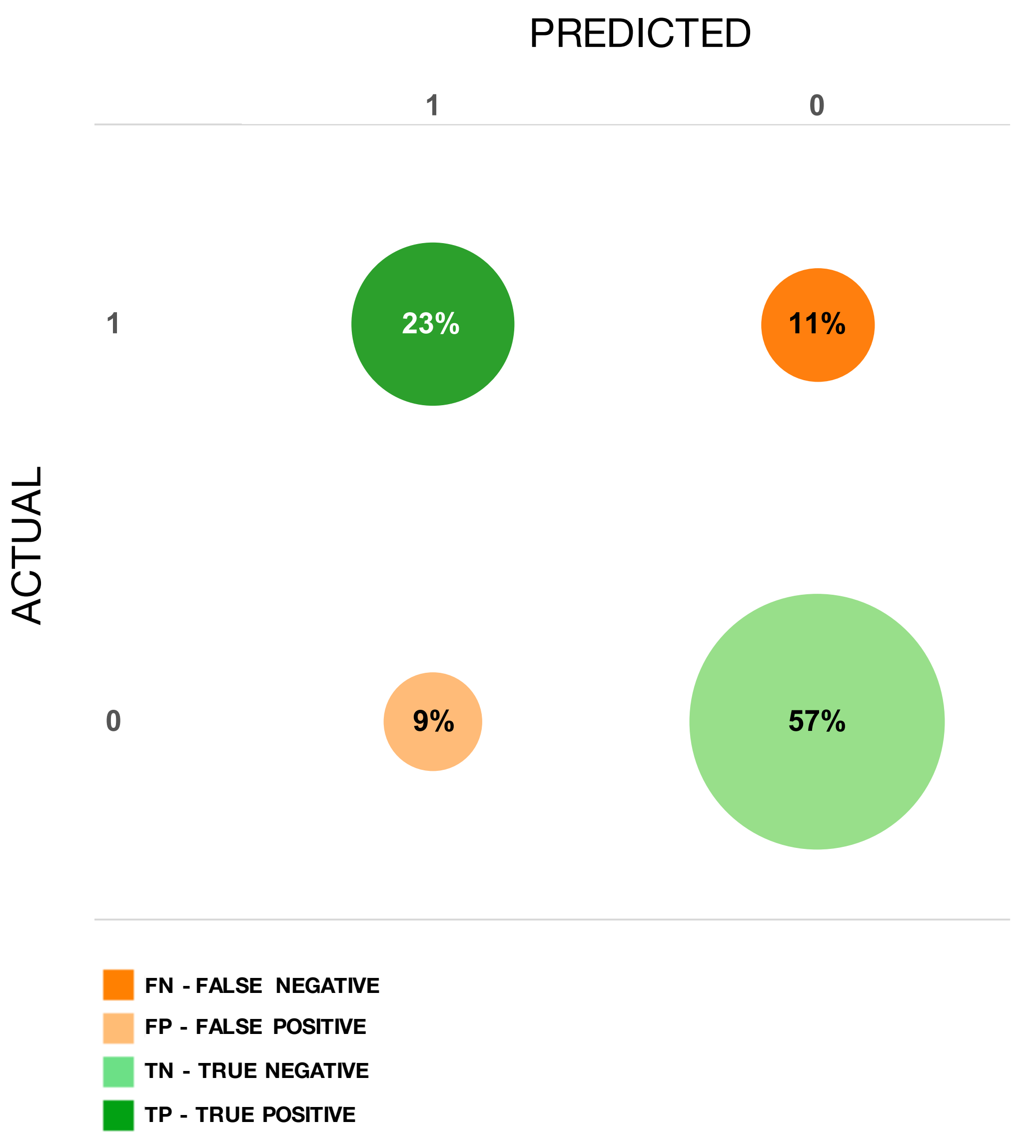
M1 on full road, 15 minutes - accidents and vehicle problems



S	Test area: 48km
T	15 minutes
I	Accidents and vehicle problems

The quality of prediction is relatively stable over the week days, with slightly better predictions over the weekend.

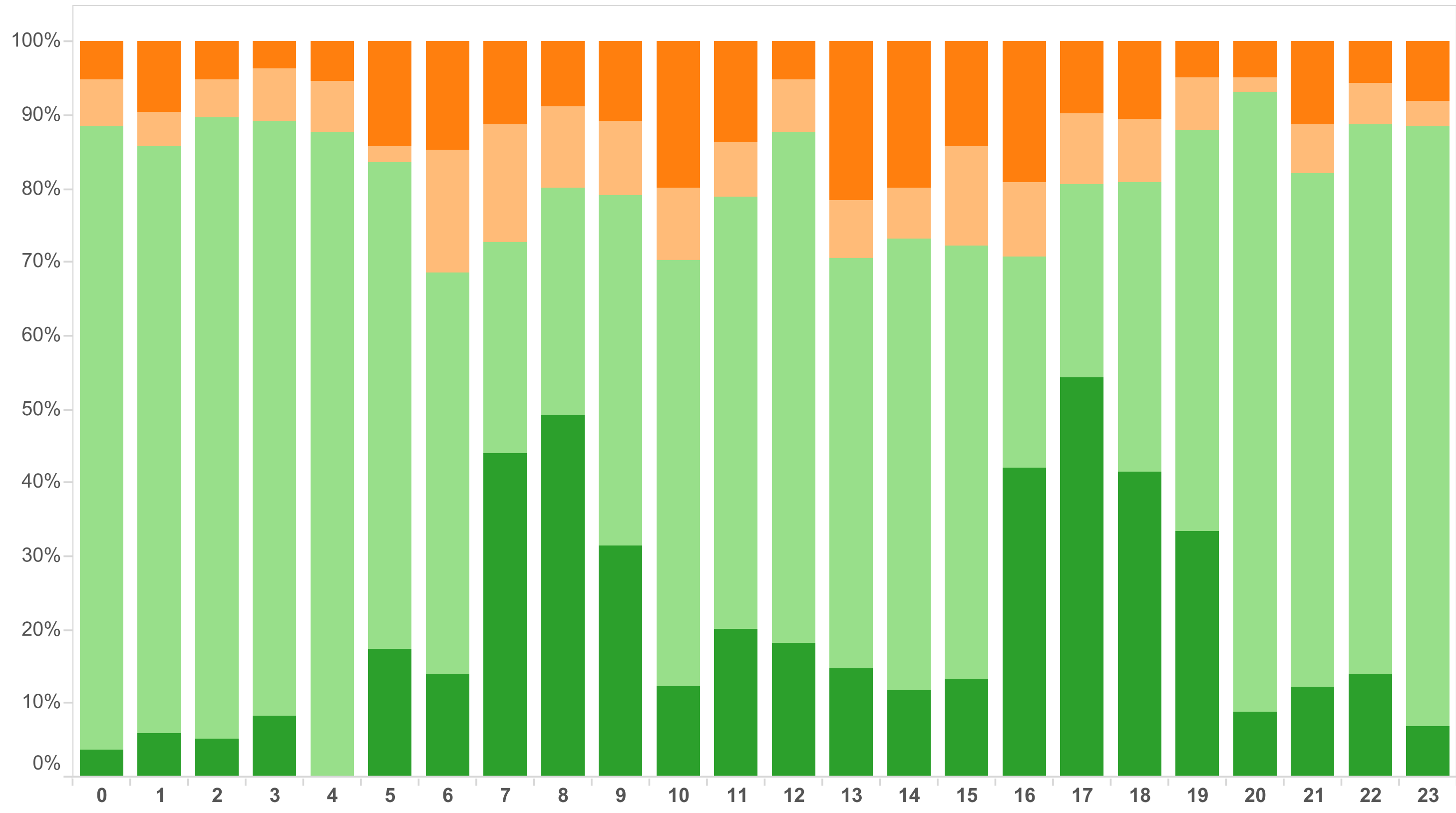
M1 - FULL ROAD 15 minutes - accidents only



S	Test area: 48km
T	15 minutes
I	Accidents

The network predicts well both true positives and trues negatives and makes similar errors on the false positives-negatives.

M1 - FULL ROAD 15 minutes - accidents only



FN - FALSE NEGATIVE

FP - FALSE POSITIVE

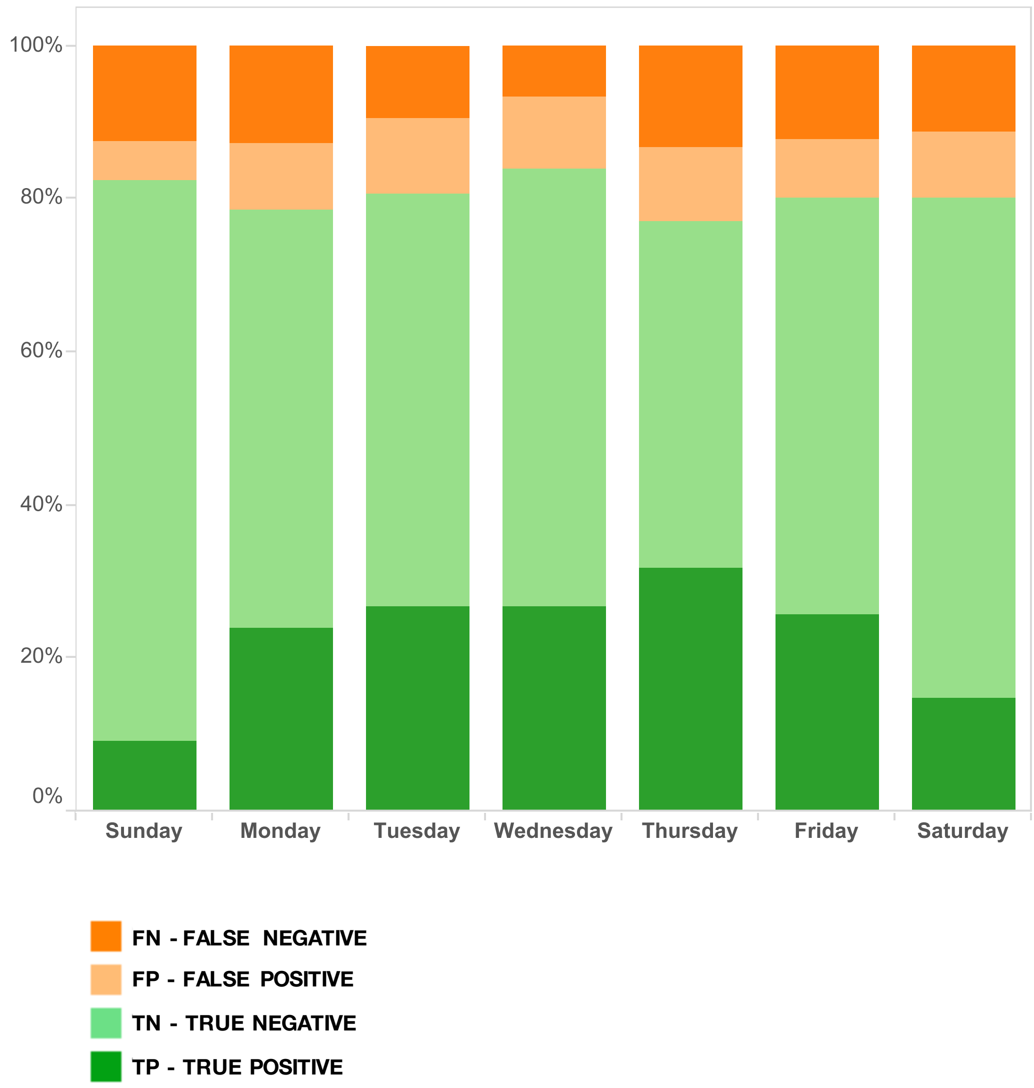
TN - TRUE NEGATIVE

TP - TRUE POSITIVE

S	Test area: 48km
T	15 minutes
I	Accidents

The network predicts well at all times of the day, including night hours when the incident occurrence is small.

M1 - FULL ROAD 15 minutes - accidents only



S	Test area: 48km
T	15 minutes
I	Accidents

The network predicts in a consistent way for all days of the week, with minor differences between week and weekends.

6

CONCLUSIONS AND RECCOMMENDATIONS

Summary

- The project has tested three Neural Network (NN) designs applied to the prediction of incidents on the Dutch highways.
- With the term “prediction” we mean predicting the status of a road section (S) in a time window (T) for a type of incident (I). With “status” we mean “Incident” or “no incident”.
- Traffic data, measured by highway loops, has been used as the sole predictor of incidents.
- The NNs designs are tested on a section of the A12 south of Utrecht. The main design variables included:
  - S: Different lengths of the test road, from full road to parts of it
  - T: Different times windows in the future (from 15 minutes to 2 hours)
  - I: Different compositions of incidents
- The performance of each NN is compared against the Binary Strategy, which is a prediction based only on past statistics of incident data.
- The project confirms that algorithms trained on past data are capable of increasing the predictability of incidents above pure statistics based on history.
- The networks perform best on the longer road section and longer time windows. Performance tends to degrade reducing road length and time window (this can be attributed to limitations of the NN design or to insufficient training).
- Data issues related to incomplete traffic data records have impacted the ability of reconstructing coherent and complete traffic time-series locally. This made it hard to train a NN for short road sections and therefore to test this case.

Observations on predicting different types of incidents

- NNs are capable of predicting all types of incidents tested. This is somewhat surprising since a variety of incidents should have limited or no correlation with traffic. More specifically:
  - All tests show that NNs can predict all types of incidents at least for long time windows and long road sections
  - M1 indicates that the prediction improves rapidly when predicting “vehicle problems+car accidents” and in particular “car accidents” alone.
- The likely explanation is as follows:
  - All NNs, with different strategies, look at which travel patterns correspond to an incident occurrence.
  - Since vehicle problems and car accidents together account for the largest portion of incidents, it follows that most incidents are strongly (car accidents) or weakly (vehicle problems) associated to traffic features, such as speed and flow.
  - When predicting all incidents without distinction, good predictions mix with other predictions and up to reasonable overall prediction.
  - The tests performed also indicate a strong improvement in predictability for car accidents alone, which is plausible given the strong qualitative link between accidents and traffic.

Review of hypothesis

- **H1:** For at least some S-T combinations, NNs can learn to reproduce the baseline incident probability for all type of incidents (not only road accidents).

The tests support this hypothesis. The tests for short road sections however are not conclusive yet because of data problems that have prevented a proper network training for short sections.

- **H2:** For incidents for which there is a known qualitative relationship between data and incident, NNs are capable of predicting above baseline for multiple S-T-I combinations.

- **H3**: higher prediction accuracy is associated to a stronger qualitative relationship between influencing factors and incident.

The tests carried out with M1 show that road accidents can be predicted with as much as 82% accuracy on 15m time window for the long road section. This is a single test but it strongly favors both hypothesis. If confirmed with additional tests, this indicates that predicting road accidents and other classes of incidents can be potentially achieved with high accuracy.

Observations on different NN strategies and their training

- Convolutional networks perform well and improve the results compared to fully connected NN. This is the case for 10m data and for 1m data.
- The increase in accuracy compared to the Binary Strategy is significant for medium to long sections of the road. The performance however degrades at shorter time intervals but especially for shorter stretched of the road. As the road sections gets shorter, the networks tested tend to produce average predictions.
- This can be the result of multiple factors:
  - The network design is not appropriate for short road sections and local predictions
  - The training and network tuning was too limited to achieve good outcomes
  - At local level, traffic alone is not sufficient to predict incidents
- Recurrent NN are, on paper, the most suitable networks for short sections. However, the difficulty in training the network, caused by data gaps, made the tests so far inconclusive.
- NN are trained on historical data and are optimized for a variety of parameters, which greatly influence the overall NN performance. They influence the way the NN learns, the sensitivity to data variations, the speed of learning etc. From experience, finding the right configuration is important for the outcome. It is also a non trivial task (think of this as tuning a radio receiver to capture a radio signal of unknown frequency).
- Selecting the right NN design parameters is a specific optimization effort, which has been done is a very very limited manner in this project. The implication is that several prediction improvements have not yet been realized.

Observations on data completeness and consistency

There are several issues with the traffic data that impact the ability of training the NN:

- The data is incomplete at every single point in time
- Complete data sequences for e.g. 1hour for clusters of sequential loops are few
- The loops change ID, location and status over time (on-off)
- The dynamics of the road is such that changes in number of active lanes is rather frequent

This results into:

- A much smaller number of viable data sets to train the networks, which impacts especially short sections and short time frames
- A high degree of noise in the data, which confuses the network and degrades predictions
- A significant learning curve to detect patterns in the data, which would otherwise be shorter

There are also issues with the incident data:

- The classification of incidents changes over time as well as the incident descriptions. At a high level of aggregation this is not an issue, but it could indicate misclassification.
- The training of the NN assumes that all incidents are uncorrelated events. It is unclear if this is the case or if some incidents are classified in multiple groups.
- The training assumes that the incident start time recorded in the incident reports is correct and reflects the time when the event occurs. If the difference between time of event and time of registration is large, then the NNs are trained with noisy data, reducing the quality of predictions.

Next steps

There are many options to develop this further, aiming at a system that assists traffic managers dealing with incidents and above all, affecting the chance that they occur.

This is a high-impact exercise with many and consequences, and it requires multiple iterations to bring the full results to fruition. From the current stage, the following are the most immediate opportunities for improving results:

- Data modeling: improving data quality through interpolation/imputation and reducing ambiguity/sensitivity on incident classification and time registrations
- Prediction strategy: spelling out which NN works best for which S-T combination and which network parameters perform best
- Model enhancements: adding high-resolution weather and environmental conditions
- Scaling: testing the models on different areas and roads to develop a possible scaling strategy
- Optimize technology: to compare performance and costs of various cloud solutions and NN software

The following table illustrates the suggested next steps.

CSRESEARCH

Suggested next steps

Data modeling	Reducing gaps and reducing dependence to changes in road layout: • Data interpolation and imputation • Data generalization • Incident better specification • Visualization of data
Prediction strategy	Establishing the relationship between NN types a the prediction canvass • "Optimal" network structure for the prediction canvass including hotspots • "Optimal" prediction parameters (road section length, time advance, network hyper-parameters)
Model enhancements	Maximize use of current data, add inputs to increase prediction quality: • Slippery roads, light • High-resolution weather • Lane disambiguation
Scaling	Explore options and implications to produce large-scale predictions • Test on different road sections (A12 and A10) • Develop hypothesis for national scaling
Optimize technology	Assess implementation options to reduce training time, increase scalability and minimize costs: • NN frameworks • Training strategies (CPU, GPU) • Cloud options
Test environment (optional, possibly for a later stage)	Design a test environment to produce real-time incident predictions • Design the real-time feed and data extraction • Design a basic GUI to access and display prediction data • Identify technical infrastructure choices

A

APPENDIX

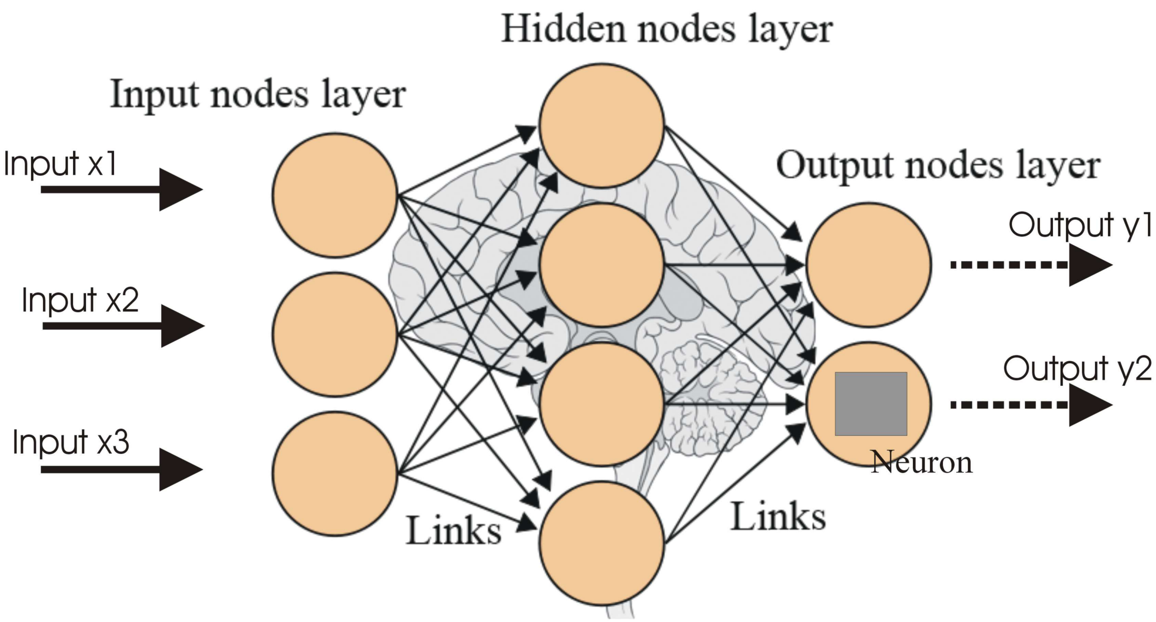
CSRESEARCH

Infrastructure

All computations and data processing have been carried out on an Amazon Web Services (AWS) infrastructure. The resources used are as follows:

- General computing for data processing
  - Data is ingested in AWS S3 (storage): 1TB
  - Initial processing and filtering is carried out in Python on AWS EC2: 8TB temporary disk
  - Additional processing is carried out in AWS Redshift (data warehouse)
  - The transformation of data into smaller training sets of images/time series in done in EC2 (Python)
- The neural networks are trained in AWS EC2
  - 32 cores, 244 GB RAM, 1TB Disk
  - The neural networks are based on code from Python, Theano and Tensorflow
- Processing time
  - The full set of set of tests required 420 hours processing time.
  - The vast majority of the processing time was taken up to cycle across neural network hyper-parameters. Once selected, the network trains in a few hours.

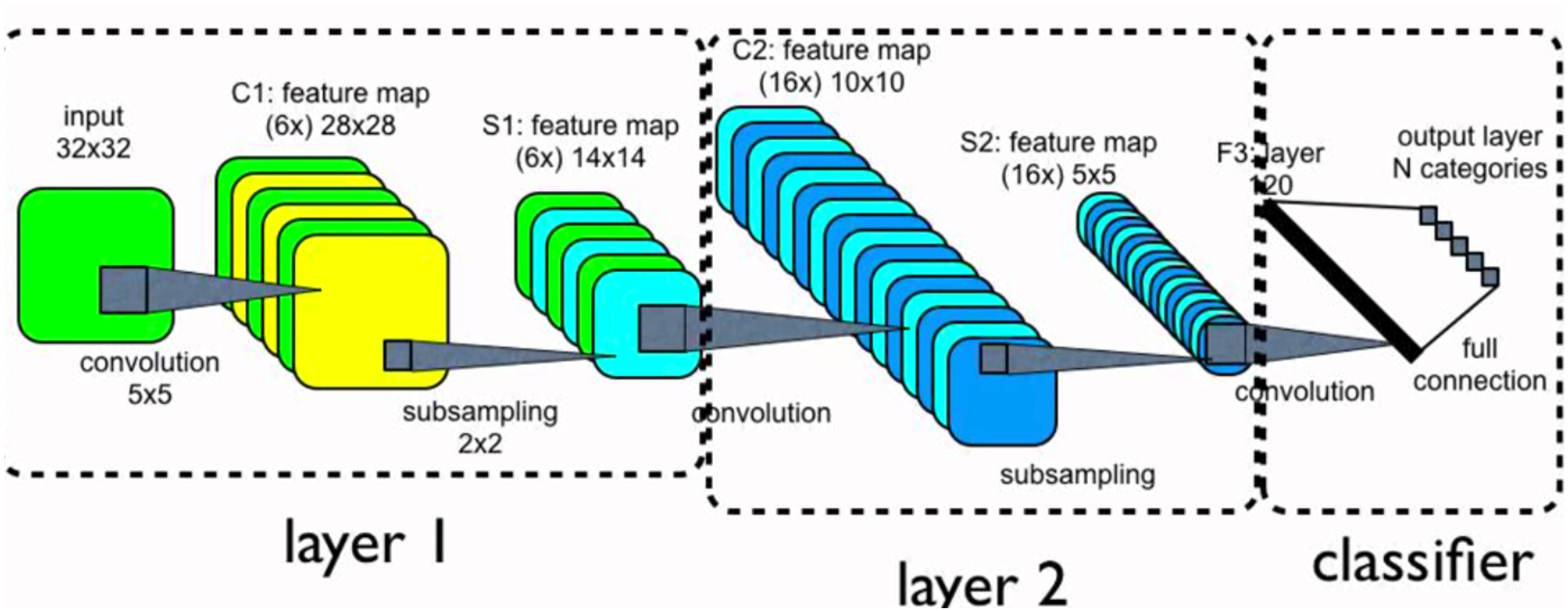
FULLY CONNECTED NEURAL NETWORKS



- Neural Networks are machine learning algorithms used for prediction and classification. Their structure is inspired by basic connectivity features of biological neurons.
- They were first explored widely in the ’80s and ’90s. Crucial breakthroughs in the mid ‘00s led to rapid development, and neural nets are now achieving record accuracies in important classification tasks like image and voice recognition.
- A neural network consists of layers of “neurons” (i.e. variables that take on real values) connected in pairs between one layer and the next by weights. It is usually represented by a graph where neurons are the nodes and weights are the edges connecting them.
- The input layer of nodes takes in the data values. Each arrow represents a weight. When a weight is large, the value of the node at the tip of the arrow is very sensitive to the value of the node at the base of the arrow. A small weight means the two nodes are insensitive to each other.
- The values at the output nodes for a specific input represent the network’s prediction. For example, with two nodes at the output representing two possible future states (e.g. incident/no incident), the prediction of the network is the state corresponding to the node with the higher value.

CONVOLUTIONAL NEURAL NETWORKS

- Convolutional networks are a special flavour of Neural Networks that perform extremely well for image, voice and text recognition.
- In a traditional neural network each input neuron connect to each output neuron in the next layer. If the number of input neurons is very large the resulting network may quickly become too large to be practical.
- In convolutional network, the input layer is split into smaller portions which are connected to a specific neuron in a feature map. Several feature maps are computed from the same input, to detect different types of features (in an image they may be edges, shapes, blobs of color etc.).
- This process is repeated multiple times (multiple layers) until the an output layer of classification. For instance, if the network is trained to detect types of animals in the input images, the output payer may contain one neuron per type of animal that we want to detect.
- The networks are designed to detect input patterns that repeat over the entire input and to specialize each layer into some specific form of pattern detection.



RECURRENT NEURAL NETWORKS

- Recurrent Neural Networks (RNNs) are types of neural networks specifically designed to process data and provide predictions *sequentially in time.*
- The nodes in RNNs are connected in a manner that allows them to retain internal memory of past data points, and decide the prediction based on learned patterns of their succession. In the image below, Xn are the inputs, provided to the network A sequentially and hn denote the corresponding outputs, also provided sequentially in time. The network A retains a sort of internal memory of past data points, which is passed to the next step. The blue highlights indicate that a certain output may be influenced from data points retained in the memory.
- The most commonly used type of RNN architecture is the so-called Long Short-Term Memory (LSTM). LSTM networks have shown high rates of success in tasks such as voice recognition, handwriting recognition, time series prediction, robotics and protein analysis. Recent advances in the architecture of LSTM networks (grid LSTM) have shown potential for further improvements in accuracy and new applications.

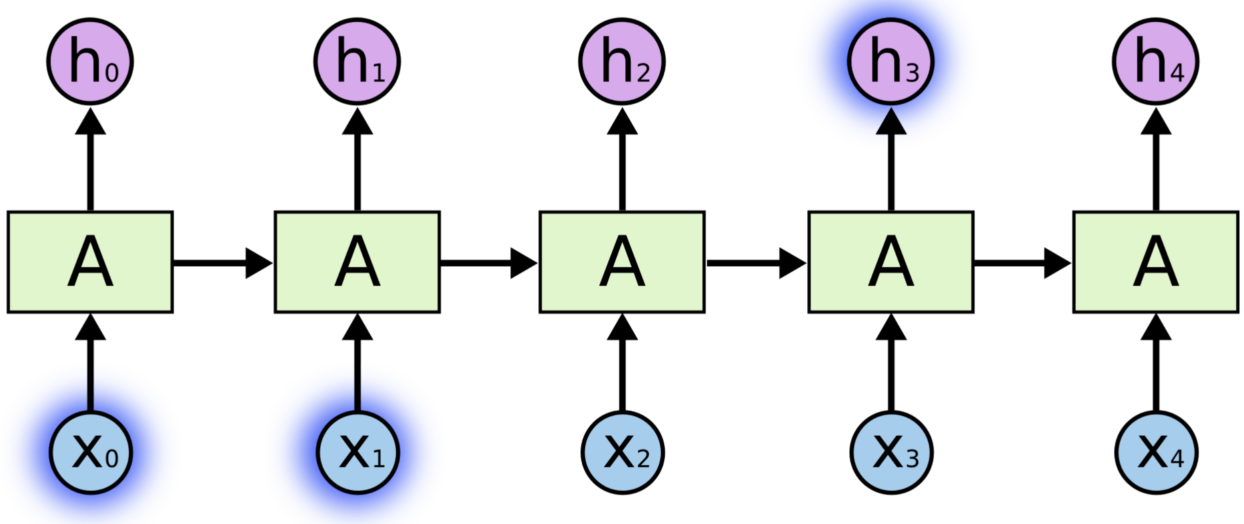
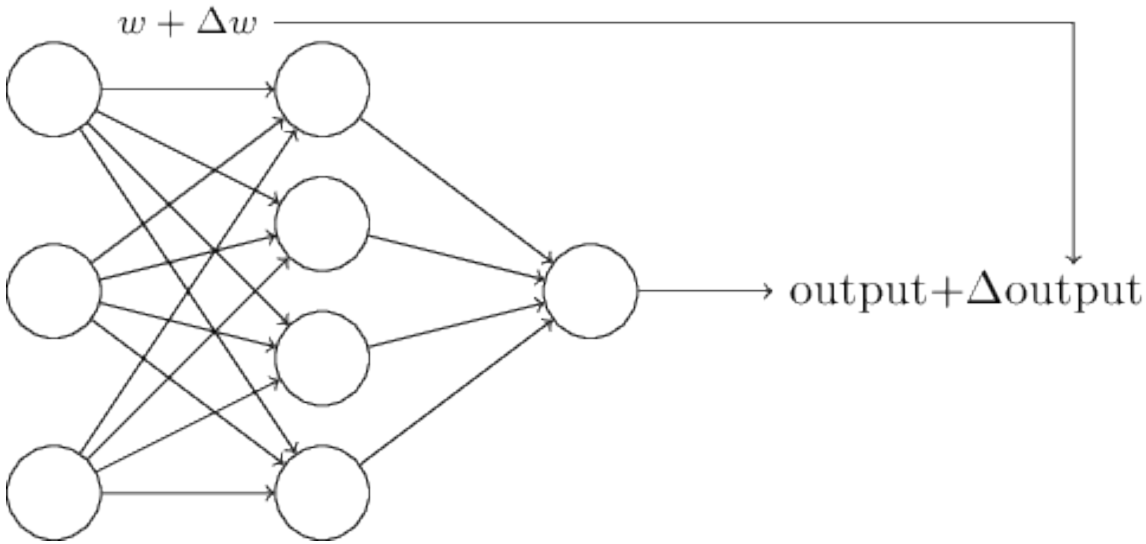


Image source: http://colah.github.io/

TRAINING A NEURAL NETWORK: BACKPROPAGATION

- During training, the values of the output neurons are compared to the actual values in the training data. The difference of the output values with expected (real) values is *backpropagated* through the network and weights are adjusted to reduce the error.
- *Backpropagation* is a technical term, referring to the precise manner adjustments are made to the weights when the network makes a wrong prediction. Neural nets first attracted interest as *universal approximators,* algorithms able to approximate any functional relationship between input and output. In practice however neural networks with many hidden layers proved hard to train. It was advances in backpropagation after the mid-'00s that allowed the training of many layers and the achievement of record accuracies in demanding machine learning tasks, especially image recognition.



Selected references

M. A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2014.

C. Olah, "Understanding LSTM Networks", 2015

Y. Bengio, I. Goodfellow, A. Courville. "Deep Learning". MIT Press, 2015.

CUDA Deep Learning frameworks https://developer.nvidia.com/deep-learning-frameworks

Google TensorFlow documentation

M. G. Karlaftis, E I Vlahogianni, "*Statistical methods versus neural networks in transportation research: Differences, similarities and some insights*", Transportation Research Part C: Emerging Technologies, 19, issue 3, 2011.

S. Roshandel, "*Impact of real-time traffic characteristics on freeway crash occurrence : systematic review and meta-analysis*", Masters by Research thesis, Queensland University of Technology, 2014.