Center for
Big Data Statistics

Working paper

Inferring network traffic from sensors without a sampling design

Working paper no.: 02-21

Jonas Klingwort
Joep Burger

May 2021

Contents

Abstract

The use of non-probability data as a primary data source in official statistics is currently an active field of research. Without the traditional sampling design, modern machine-learning algorithms might play a central role in producing accurate population estimates. This working paper presents empirical research on the effects of class imbalance and the non-probability nature of the data on the quality of individual predictions and population estimates.

Using a graph-theoretical interpretation of the Dutch state road network, with traffic junctions as vertices and state roads as edges, the Dutch freight traffic across network is inferred from Weigh-in-Motion (WiM) road sensors. These sensors are installed on a non-probability sample of edges detecting passing transport vehicles. Photographs of the license plates allow for record linkage of vehicle features. However, the sensors only provide information about a small and non-probability sample of edges in the network. Another complication is that detecting a vehicle from the population (the vehicle register) is a rare event. We apply extreme gradient boosting to learn the probability of vehicle detection by a WiM sensor from time features (e.g., weekend indicator, weather), edge features (e.g., pageRank of an edge's origin vertex, general traffic intensity from loop sensors), vehicle features (e.g., mass, age) and vehicle owner features (e.g. company size, economic activity). The learned relationship is then used to predict the probability of detection on each day of the year, along each edge in the network for each vehicle in the population. Several scenarios were designed to simulate the effects of the non-probability nature of the data and of the extreme class imbalance.

With about 27 million records and over 100 features, the model performed about halfway between random guessing and perfect prediction when trained and tested on a balanced probability sample. Training and testing on a non-probability sample caused substantial variation in model performance across test sets, confirming the risk of extrapolating to domains that are not well represented in the data. Class imbalance seriously compromised model performance, best detected by Matthews' correlation coefficient and the min-max normalized F_1 of the rare class. Balancing the data improves model performance on balanced test sets but hampers making inference to the entire population, illustrated by reliability plots.

Producing official statistics using non-probability data as the primary source would benefit from a sampling design or features explaining the data generating mechanism. In the absence of both, and when predicting a rare event, the quality achieved with a modern machine-learning algorithm does not meet official statistics' quality standards.

Keywords— weighted directed graph, non-probability sample, road transport network, freight traffic, Weigh-in-Motion road sensor, extreme gradient boosting

1 Introduction

In official statistics, big data potentially allows to produce statistics cheaper, faster, and on a higher level of detail (Daas et al., 2015). Big data in context of this paper is data recorded by sensors and not based on a sample survey or an administrative source. Sensors are assumed to be more accurate, reliable, and faster than sample surveys (Hackl, 2016). In contrast to traditional sample surveys, however, big data typically lacks a sampling design: not every element in the population has a known and positive probability of being observed. The population coverage is not only incomplete but also unknown. Without a sampling design and a known data generating mechanism, no valid inferences can be drawn by design-based approaches. Correcting for missing data using design-based inference methods is therefore impossible. Accordingly, new methodology is needed to use big data as the primary data source for population inference in official statistics (Buelens et al., 2018).

In this study, we infer the Dutch freight traffic across the Dutch road transport network using data from road sensors installed on a non-probability sample of road segments. The road sensors detect road freight vehicles, but the sensors were not installed to infer the freight traffic distribution. Not all road segments of the transport network have sensors installed, and road segments with sensors installed have not been randomly sampled. Can we make inference from the non-probability sample to the population? To improve readability, the word 'vehicle' is used here as a synonym for 'road freight vehicle', thus excluding bicycles, cars, trains, ships, planes, etc. Most road freight vehicles are trucks and tractors.

To achieve this study's aim, we represent the road network as a graph, with traffic junctions as vertices and state roads as edges. In the first step, gradient boosting is used to model the probability of detecting a vehicle by a sensor on a state road segment as a function of time, edge, vehicle, and vehicle owner features. In the second step, the learned relationship between features and vehicle detection is used to predict the detection probability for any vehicle in the vehicle register on any day along any edge in the road network with or without a sensor. Traffic intensities can be derived by adding these probabilities for all vehicles or certain vehicle classes. Elliott and Valliant (2017) refer to this alternative to design-based approach as a superpopulation modeling approach.

Like many other datasets, the sensor data is severely imbalanced, i.e., most vehicles in the license plate register do not pass any sensor in the time frame covered. To study the effect of this class imbalance on model performance, we train models on both random and balanced samples of the data. To study the data-generating mechanism's effect, we also simulate both probability and non-probability partitioning of the data into training and test sets. The model prediction quality for the different scenarios is assessed using several performance metrics and K-fold cross-validation.

We will show and discuss issues encountered during this project concerning selectivity and imbalanced data, which are highly important for population inference and official statistics. We will demonstrate potential solutions and derive recommendations for the implementation of the methodology in practice. Finally, we will show that the proposed methodology can infer and visualize the Dutch transport vehicle traffic distribution in the state road network throughout an entire year.

The working paper is structured as follows: in Section 2, we address the research background, all data sources are described in Section 3, the methodology is explained in Section 4, the results are shown in Section 5, a discussion is given in Section 6. The paper concludes with Section 7.

2 Research background

Methodology to implement big data in statistical production processes in official statistics is an actual and ongoing research topic (Baker et al., 2013; Kim et al., 2014; Tam & Clarke, 2015; Shlomo & Goldstein, 2015; Hackl, 2016).[1] There are many similarities to survey research using non-probability data, as big

[1] See also ESSnet Big Data I and ESSnet Big Data II for the integration of big data in the regular production of official statistics (`https://ec.europa.eu/eurostat/cros/content/essnet-big-data-1_en`).

data are typically non-probability samples. Especially, how to utilize non-probability data for population inference is an ongoing research topic (Elliott & Valliant, 2017; Vaillant, 2020). In general, the combination of non-probability with probability data has proven to be a promising approach (Lohr & Raghunathan, 2017; Klingwort, 2020). A recent review on conceptual approaches for non-probability research is given by Cornesse et al. (2020).

Big data can be used either as auxiliary information within established methodology or as a primary data source (De Broe et al., 2020). Using big data as auxiliary information in model-based inference is generally more straightforward and similar to using auxiliary information from population registers. In such a scenario, the models' quality with and without big data can be assessed. Although combining non-probability data with probability data is a promising approach, probability survey samples remain expensive, slow, and burdensome. In this study, we, therefore, aim to obtain population inference using non-probability data only. As mentioned in Section 1, the data to be used lacks a sampling design, principles of probability sampling are omitted, and therefore, without developing an inference framework, no valid inferences can be drawn. This challenge is commonly encountered when using 'found data' (Harford, 2014) for population inference.

Previous work on this project can be found at Klingwort et al. (2019b), where a simpler model predicting counts was used instead of predicting individual probabilities as will be done in this study. This methodological development is possible because the sensors have cameras for automatic license plate recognition, allowing linkage of population register data at the micro-level using license plate as the linkage key. In this case, the term big data can be expanded to 'identifiable big data' (Shlomo & Goldstein, 2015). By linking big data observations to a population register, auxiliary information from the (governmental) register is provided for the big data observations, which is often impossible in practice (Buelens et al., 2014; Schnell, 2019). Furthermore, all available datasets (Weigh-in-Motion sensor data, traffic loop sensor data, vehicle and entperise register data) were linked, which were previously researched separately or partially combined in transport and mobility research at Statistics Netherlands (Daas et al., 2015; Puts et al., 2019; Klingwort et al., 2019a; Gootzen et al., 2020). Hence, a part of the concept of combining datasets is used, but not in the sense of combining non-probability with probability-based data. Instead, it is used by enhancing the big data source with auxiliary information. This study design will allow evaluating the prediction's quality and accuracy when using non-probability data as the primary source.

Our proposed methodology is based upon a network analysis. Statistics Netherlands is already applying such network analysis (Buiten et al., 2018). van der Laan (2019) used a social network to measure segregation in the Netherlands. Here, the network traffic in the Dutch state road network is modeled. We interpret the Dutch state road network as a graph, with the traffic junctions being the vertices and the connecting state roads the edges. The vertices represent the actors of the network, and the edges represent the connection between those. Specific details on the constructed graph used in this application are given in Section 3.1, while we recommend Jungnickel (2005), Hsu and Lin (2009), van Steen (2010) for further reading on the topic of network and graph analysis.

Finally, concerning the inference method, we base our study on algorithmic inference. Here, modeling is seen from a different perspective compared to model-based approaches (Breiman, 2001). Instead of fitting a model, an algorithm is tuned and optimized concerning predictive power (Buelens et al., 2012). Although model-based inference would be feasible as well, the supervised machine learning algorithm scales easier with the number of available features (see Appendix A). We will explain this in more detail in Section 4.

3 Data

The transport network is described in Section 3.1. The sensors of a Weigh-In-Motion road sensor network recording transport vehicle traffic are described in Section 3.2. The target variable is defined and described in Section 3.1. The features used for prediction are subsequently described in Sections 3.4 (time features), 3.5 (edge features) and 3.6 (vehicle and owner features).

3.1 Transport network

The Dutch freeway network information was obtained by scraping interchange road junctions and their connecting freeways from `https://www.wegenwiki.nl`. For scraping, the R-library `rvest` was used (Wickham, 2019). The Dutch transport network of freeways was constructed as a graph, using the interchange road junctions as vertices and their connecting freeways as directed edges. To construct the graph, the library `igraph` was used (Csardi & Nepusz, 2006). For realistic vertex feature values, the freeway network was expanded with neighboring freeways in Belgium and the German federal states North Rhine-Westphalia, Lower Saxony, and Bremen, which share a border with the Netherlands. For population inference, only the Dutch network will be used. Figure 1 shows the resulting network. The Dutch part consists of 108 vertices and 286 edges. The 18 road sensors (see Section 3.2) were mapped to 18 edges of the network using their geographical location. The transport network is represented by a weighted directed graph consisting of V vertices (traffic junctions) and $E = V(V-1)$ potential edges (state roads). Note, the $V \times V$ adjacency matrix is sparse, i.e., it contains mostly 0s and only 286 1s. The graph is represented by a $V \times V$ adjacency matrix W with w_{od} the weight of the edge from origin vertex o to destination vertex d.

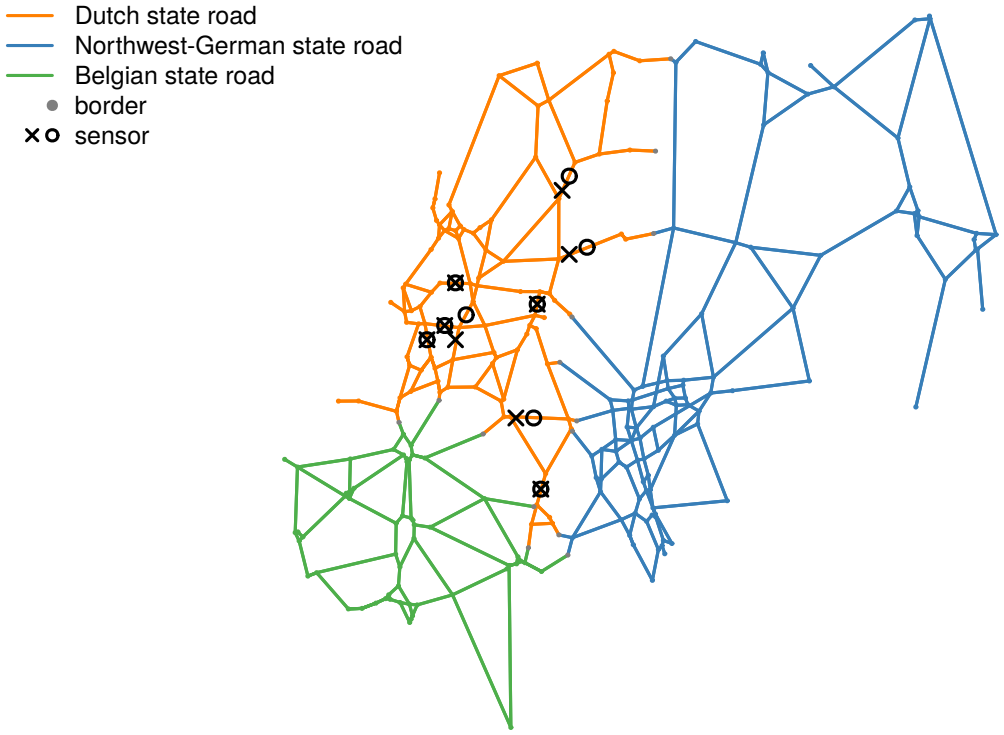


Figure 1: The transport network in geographical layout: Dutch (orange), Belgian (green), and North-West German (blue) interchanges, connected by freeways. Grey vertices are border crossings. Black crosses and circles indicate the two stations of the nine WiM systems.

3.2 Weigh-in-Motion road sensors

The data collected by Weigh-in-Motion (WiM) sensors is commonly used in '[...] bridge structural analysis, geometric design, safety analysis, traffic control and operations, freight management, and operations, facility planning and programming, and standards and policy development' (Federal Highway Administration, 2007, page 1). In 2015 nine WiM systems were operating on Dutch freeways. Each system is split into two separate stations recording traffic in both directions (see Figure 1). For simplicity, here the network is shown as an undirected graph. The sensors are installed on every potential lane of the particular freeways (potential lanes are: hard shoulder, slow, intermediate, and fast lane), resulting in recordings on 64 lanes. However, cameras identifying license plates were operating only on slow and intermediate lanes, reducing the number of lanes eligible for analysis to 36. Fortunately, the lanes with cameras are also the lanes used most by transport vehicles. Thus, within the weighted directed graph, 18 out of 286 edges in our network have WiM sensors.

When a vehicle passes a sensor station, it is weighed, classified, timestamped, and a photograph of the front license plate is taken. An example recording is shown in Figure 2. The photo taken shows the entire vehicle. The number of recorded axles and their approximate weight is shown below the photograph using numbered bars for each axle. The exact axle measurements and the distance between the axles are given in the right panel. Moreover, on the upper right side, the license plates (front/rear), the country code, the sensor's id passed, and the driven speed are shown. The recorded license plate is not shown here due to data protection. The timestamp (upper left corner) and automated vehicle classification (lower left corner) are included within the photo. Optical character recognition (OCR) of the front license plate (not shown here) and the time stamp provide a unique identifier to link WiM recordings to register information on the micro-level. This opportunity allows filtering on vehicles that belong to the considered population and linking vehicle and owner features from the vehicle and business registers. The considered population is the Dutch commercial vehicle fleet, excluding military, agricultural, and commercial vehicles older than 25 years, with a weight of at least 3.5 t (empty vehicle weight + loading capacity). This is the definition used in the Dutch Road Freight and Transport Survey (Centraal Bureau voor de Statistiek, 2017). Fortunately, WiM sensors also record only vehicles from 3.5t. More information on this specific dataset is given by Klingwort (2020).

Figure 2: WiM software showing sensors recording a passing truck.

3.3 Target variable

The number of target variables using WiM data is limited. One option is to define the target variable as the transported shipment weight (in kt). This would require measurement error corrections (see Klingwort (2020, section 5.2) for the required analyses). Therefore, we limit this study to a simpler target variable: an indicator for vehicle detection.

We define the indicator y_{tei} being the target variable, which equals 1 if during day t along edge e transport vehicle i was recorded at least once, and 0 otherwise. The dataset to be considered for analysis contains ($T = 365$ days) $\times$ ($E = 18$ edges) $\times$ ($N_q \approx 135,000$ registered transport vehicles) ≈ 887 million rows. The subscript q indicates the quarter of the year, as the license plate register is updated quarterly. The sensors recorded a total of about 36 million transport vehicles in 2015. Of those, about 25 million vehicles had a recognized license plate. About 15 million could be linked to the Dutch license plate register because they belong to the considered population. About 14 million recorded vehicles remained after removing duplicates per day (the same vehicle passing the same sensor multiple times a day). In this set, the class where $y_{tei} = 1$ comprises only $\alpha = \frac{14}{887} \approx 2\%$ of the data, i.e. the odds for recording are $\frac{\alpha}{1-\alpha} \approx 0.02 \ll 1$ and the imbalance coefficient $\delta = 2\alpha - 1 \approx -0.97 \ll 0$. Class imbalance is therefore a serious issue that needs to be addressed by balancing the sample (Section 4.2), the performance metrics (Section 4.3) or both.

Figure 3 shows the daily number of vehicles recorded per system and station ($Y_{te} = \sum_{i=1}^{N_q} y_{tei}$) in 2015. The occurring stable patterns show differences between busy weekdays and quiet weekends. Time features

will take this into account (Section 3.4). However, suspicious low counts and recording gaps are observed. Especially RW28L and RW50R show large intervals with low counts or huge gaps. Reasons for small gaps were mostly due to maintenance, while the larger gaps were due to the system's malfunctions and general quality issues.

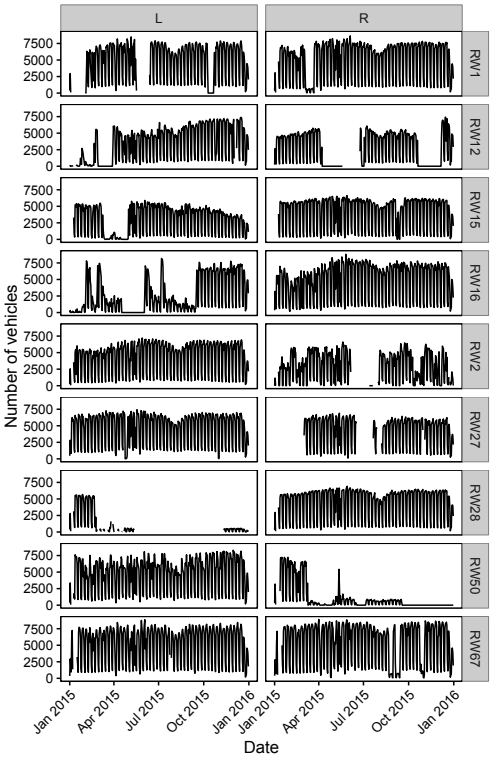


Figure 3: Number of recorded vehicles per day in 2015 split by WiM system and station. Gaps indicate a station was turned off. Flat lines indicate a station being switched on but not recording.

Severely imbalanced data can cause algorithms to fail to learn from data, and consequently, the quality of the predictions for the minority class is poor. However, the minority class is often of particular interest, for example, fraudulent transactions, network intrusion detection, and medical diagnostics (Chawla et al., 2003). Handling of imbalanced data is still researched and requires specific practical solutions for each use case. Section 4.2 will explain the chosen methods to tackle this issue in the current application.

3.4 Time features

Time features include a weekend indicator, indicators for each day of the week, and daily weather variables (Appendix A, Table 4). Weather variables were downloaded from the Koninklijk Nederlands Meterologisch Instituut (KNMI).[2] Five weather stations were selected because their temperature series are homogenized, and they contained virtually no missing values: De Kooy (NW), Eelde (NE), De Bilt (center), Vlissingen (SW), and Maastricht (SE). Their measurements, such as wind, visibility, and temperature, were averaged per day across weather stations.

3.5 Edge features

Edge features include the edge weight, network centrality measures of an edge's origin and destination vertex, and daily traffic intensity measures (Appendix A, Table 5). These features were calculated, using functions of the `igraph` library (Csardi & Nepusz, 2006). Traffic intensity is measured by loop sensors that

[2] `http://projects.knmi.nl/klimatologie/daggegevens/selectie.cgi`

are far more abundant than WiM sensors but have no camera for license plate identification and only produce counts (National Data Warehouse for Traffic Information, 2016). Although some loop sensors distinguish between small, medium, and large vehicles (in m), only the total count was used to include all loop sensors, i.e., about 12 thousand along the state roads in the constructed state road network. Raw data have been processed as described by Puts et al. (2019). Loop sensors were mapped to edges as described in Appendix B and their recordings were summarized per day and edge.

3.6 Vehicle and owner features

Register features include several technical and non-technical vehicle features from the national license plate register, such as age, mass, and type, and owner features from the business register, such as economic activity, size, and location (Appendix A, Table 6). The features are linked on a micro-level to the WiM observations using license plate and annual quarter as the unique linkage key.

4 Methods

In this chapter, we describe the algorithmic inference method (Section 4.1), the scenarios (Section 4.2), the performance metrics (Section 4.3), and how the freight traffic across the transport network is inferred (Section 4.4).

4.1 Gradient boosting

Gradient boosting is a supervised machine learning technique for regression and classification problems (for further reading see Breiman (1997), Friedman (1999a, 1999b)). The predictions of a weak learner like a shallow decision tree are boosted by adding predictions of new weak learners that are trained on the residuals of previous predictions Hastie et al. (2009, algorithm 10.3)). Newly added predicted residuals are weighted to prevent overfitting. The weight is optimized by minimizing a loss function. In contrast to model-based inference, for example, logistic regression, these prediction models scale easier with a larger number of features. It can be considered as a more natural way to model nonlinear relationships and interactions. In a logistic regression model, the relationships and interactions would need to be specified explicitly.

Gradient boosting is applied here to learn the relationship between features and vehicle detection (a classification problem). We use the framework XGBoost developed by Chen and Guestrin (2016) and implemented in the R library `xgboost` using binary cross-entropy or log loss as objective function. A class weight for the positive class was used to account for the class imbalance. The class weights will be 1 in a balanced sample and were approximately 19 in our imbalanced sample (see section 4.2). The learning rate was set to 0.8, the maximum depth of a tree was set to 6, and the number of boosting iterations (trees) to 100. These were set after some testing with a small number of trees. Test error may improve with a lower learning rate and more trees (Hastie et al., 2009). In future research, hyperparameter values could be optimized using a systematic grid search.

4.2 Scenarios

As described in Section 3.3, the data is severely imbalanced concerning the class of recorded Dutch vehicles. As described in Section 1, the sensors are also not located along a probability sample of edges. To study the effect of imbalanced data and the non-probability nature of the data, we defined six scenarios.

The labeled dataset consists of $E = 18$ edges with a WiM sensor, $T = 365$ days and $N_q \approx 135$ thousand vehicles, i.e. over 800 million rows. Two samples from this dataset were drawn (see Table 1): one imbalanced sample (odd scenarios 1, 3, and 5) and one balanced sample (even scenarios 2, 4, and 6).

The scenarios within each sample differ in their partitioning (non-probability vs. probability) and in their test set sizes (variable vs. constant).

Table 1: Scenarios by sample balance, sample partitioning, and partition size

Scenario	Non-probability Variable size	Probability Variable size	Probability Constant size
Imbalanced	1	3	5
Balanced	2	4	6

The underlying concepts of these samples and scenarios are shown in greater detail in Figure 4. The scenario numbers from Table 1 are also shown in Figure 4. For illustration purposes, we only show a subset of three edges, two days, and a handful of vehicles of each sample. In this illustration, a 1 indicates that a vehicle was detected ($y_{tei} = 1$), a 0 that it was not. In the following, we will first describe the sampling and then the partitioning.

To achieve a balanced sample, synthetic instances of the minority class could be generated (Chawla et al., 2002). Instead, we applied undersampling of the majority class to reduce the sample size. All $N_{te}^{+} = \sum_{i=1}^{N_q} y_{tei}$ observed vehicles were included in the sample The same number was randomly sampled from the unobserved vehicles per edge and per day. Thus, the sample size per day per edge was $2N_{te}^{+}$, and the inclusion probability $\pi_{tei} = 1$ if $y_{tei} = 1$ and $\pi_{tei} = \frac{N_{te}^{+}}{N_q - N_{te}^{+}}$ if $y_{tei} = 0$. Hereby, the inclusion probabilities in this sample depend on the target variable. This makes it impossible to make inference to edges without a WiM sensor. The possibility of drawing a stratified sample based on the features is addressed the discussion in Section 6.

The imbalanced sample was obtained by sampling per day per edge the same number $2N_{te}^{+}$ at random. The inclusion probability for the imbalanced sample was therefore $\pi_{te} = \frac{2N_{te}^{+}}{N_q}$. To account for the class imbalance, class weights were set to $\frac{N}{2N^{+}}$ in the gradient boosting. The total sample size in both samples was $n = 2\sum_t \sum_e N_{te}^{+} \approx 27$ million rows. The two initial samples are schematically shown in the first column of Figure 4 and are the basis for the six scenarios. Note that the samples contain more data from days and edges where more vehicles are recorded. We will address this aspect in the discussion in Section 6.

Models are trained and tested using $K = 18$-fold cross validation: the sample is partitioned into K folds, $K - 1$ folds are used for model training and hyperparameter tuning, and 1 fold is used to test model predictions. This is repeated K times, so K models are tested on K independent test sets. The scenarios differ in the partitioning of the sample.

In the non-probability partitioning (scenarios 1 and 2), each test set $k = 1, \ldots, K$ equals an edge e. This simulates our aim of applying a model trained on the non-probability sample of 18 edges with a WiM sensor to 268 edges without a WiM sensor. By performing a $K = 18$-fold cross validation, we obtain 18 estimates of model performance when extrapolating to edges not represented in the training set.

In the probability partitioning (scenarios 3 through 6), the data are randomly distributed across the test sets. The test sets are therefore well represented in the training sets. In the probability partitioning with variable test set size (scenarios 3 and 4), each test set k contains as many vehicles as in the non-probability partitioning ($n_k = n_e = 2\sum_t N_{te}^{+}$) but in contrast to the non-probability partitioning they do not come from a single WiM sensor. Instead, a proportional allocation is applied to distribute all $2N_{te}^{+}$ vehicles across the K test sets, each test set k receiving $\frac{n_k}{n}$ of the data per day per edge. In the probability partitioning with constant test set size (scenarios 5 and 6), each test set contains the same number of observations ($n_k = \frac{n}{K}$). All $2N_{te}^{+}$ vehicles are equally distributed across the K test sets, each test set k receiving $\frac{1}{K}$ of the data per day per edge.

For all scenarios, the predicted probabilities can be compared with the observed ones, allowing to assess the quality of the model predictions. These results will be shown in Section 5.

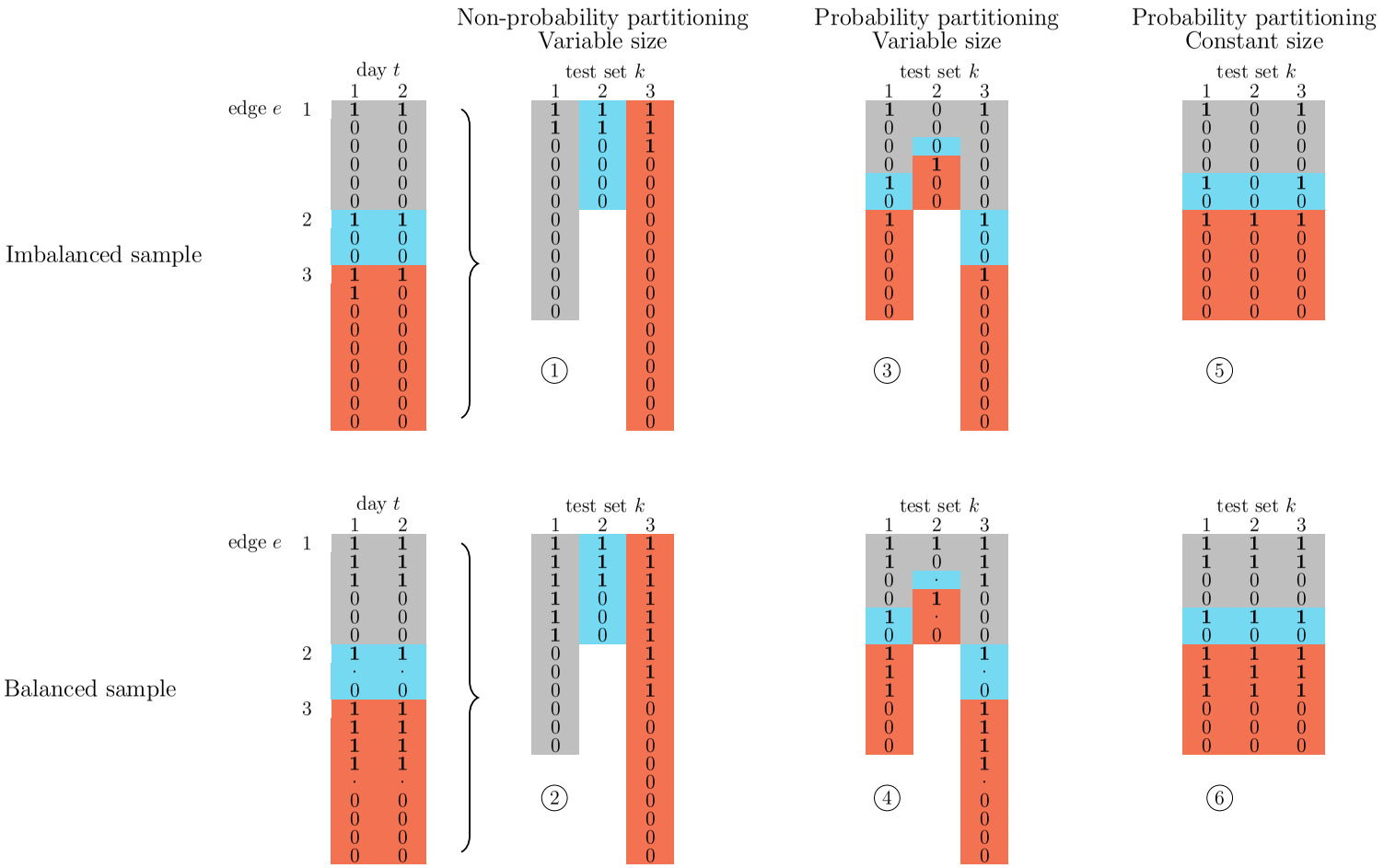


Figure 4: Scenarios by sample balance, sample partitioning, and partition size.

4.3 Performance metrics

Several well-known performance metrics were used to assess the quality of the model predictions (Powers, 2011). All are based on the confusion matrix (Table 2). All symbols refer to a single test set, but subscript k is omitted for readability. TP_c is the number of true positives at cutoff $0 \leq c \leq 1$ above which the predicted probability of recording $\hat{p}_{tei}$ is translated into label 'recorded'. FN_c is the number of false negatives, FP_c the number of false positives, TN_c the number of true negatives, n^+ the actual number of recorded vehicles, n^- the actual number of vehicles not recorded, m_c^+ the predicted number of recorded vehicles, m_c^- the predicted number of vehicles not recorded and n the sample size. Note that the actual numbers n^+ and n^- do not depend on threshold c but the predicted numbers m_c^+ and m_c^- do.

Table 2: Confusion matrix for a single test set k.

		Predicted Recorded	Not recorded	$\sum$
Actual	Recorded	TP_c	FN_c	n^+
	Not recorded	FP_c	TN_c	n^-
	$\sum$	m_c^+	m_c^-	n

Performance metrics are listed and defined in the first two columns of Table 3. The third column will be explained later. Accuracy (ACC_c) is the fraction of the sample that is predicted correctly, irrespective of class.

The true positive rate (TPR_c) alias sensitivity alias recall is the fraction of the actual number of recorded vehicles predicted correctly. The true negative rate (TNR_c) alias specificity is the fraction of the actual number of unrecorded vehicles predicted correctly. These two trade off: the lower cutoff c, the higher TPR_c but, the lower TNR_c. Plotting TPR_c against the complement of TNR_c for different values of c yields the receiver operating characteristic (ROC) curve. The area under the ROC curve (AUC) is a second performance metric. The vertical distance between the ROC curve and the diagonal is Youden's J_c alias Peirce Skill Score alias binary informedness.

The positive predictive value (PPV_c) alias precision is the fraction of the predicted number of recorded vehicles that is actually recorded. The negative predictive value (NPV_c) is the fraction of the predicted number of unrecorded vehicles that is actually not recorded. The trade-off between these two metrics is captured by markedness (MRK_c). Note that the higher cutoff c, the higher PPV_c and the more stable NPV_c (larger denominator m_c^-) but the lower NPV_c and the less stable PPV_c (smaller denominator m_c^+).

Matthews' correlation coefficient (MCC_c) alias Yule's or Pearson's ϕ is a measure of association between two binary variables.

The trade-off between TPR_c and PPV_c is captured by their harmonic mean in the positive F_1 (PF_{1c}), where subscript $\beta = 1$ means that they are weighted equally. Similarly, the trade-off between TNR_c and NPV_c is captured by their harmonic mean in the negative F_1 (NF_{1c}).

Some performance metrics are sensitive to class imbalance (Luque et al., 2019). To make metrics and scenarios comparable, we min-max normalize each metric Q using as minimum the expected score $E[Q(g)]$ when randomly guessing with probability g that the vehicle is recorded: $Q^{mmn} = \frac{Q - E[Q(g)]}{1 - E[Q(g)]}$. The min-max normalized score can be interpreted as a fraction of the way from random guessing to perfect prediction. A negative score means that the model is performing worse than random guessing. Here we choose g equal to the fraction of vehicles actually recorded $\alpha = \frac{n^+}{n}$. The expected scores $E[Q(g = \alpha)]$ are given in the third column of Table 3 (Burger & Meertens, 2020). The AUC^{mmn} equals Somer's D and ACC^{mmn} resembles Heidke Skill Score alias Cohen's κ where g is set to $\frac{m^+}{n}$. The min-max normalization has no effect on J, MRK, and MCC because their expected values are (virtually) 0 with random guessing.

All metrics except AUC depend on cutoff c. To find the optimal cutoff c^*, we first average the performance score Q_{kc} over all K test sets per cutoff c: $\overline{Q}_c = \frac{1}{K}\sum_k Q_{kc}$ and then choose the cutoff value where the average performance peaks: $c^* = \arg\max_c \overline{Q}_c$. The distribution of $Q_k(c^*)$ serves as a proxy for the quality of the predictions.

Table 3: Performance metrics and their expected value when randomly guessing with probability $g = \alpha = \frac{n^+}{n}$ that a vehicle is recorded (Burger & Meertens, 2020).

Metric Q	Definition	$E[Q(g = \alpha)]$
ACC_c	$\frac{TP_c+TN_c}{n}$	$\alpha^2 + (1-\alpha)^2$
AUC	$\int_{c=0}^{1} TPR_c dTNR_c$	$\frac{1}{2}$
J_c	$TPR_c + TNR_c - 1$	0
MRK_c	$PPV_c + NPV_c - 1$	$0 + O(\frac{1}{n^2})$
MCC_c	$\frac{TP_cTN_c - FN_cFP_c}{\sqrt{n^+n^-m_c^+m_c^-}}$	$0 + O(\frac{1}{n^2})$
PF_{1c}	$\frac{2}{\frac{1}{TPR_c}+\frac{1}{PPV_c}}$	$\alpha - \frac{1-\alpha}{4n} + O\left(\frac{1}{n^2}\right)$
NF_{1c}	$\frac{2}{\frac{1}{TNR_c}+\frac{1}{NPV_c}}$	$1 - \alpha - \frac{\alpha}{4n} + O\left(\frac{1}{n^2}\right)$

4.4 Inferring network traffic

The freight traffic across the transport network on day t can be inferred by estimating the number of vehicles $\hat{Y}_{te}$ along each edge $e = 1, \dots, 286$ with or without a WiM sensor. The model performance in scenario 1 (imbalanced sample with non-probability partitioning of variable size) is the best proxy for the quality of the inferred network traffic.

The model is retrained on the imbalanced sample including all $K = 18$ edges with a WiM sensor, i.e. without leaving one out for testing. The learned model is then used to predict the probability of detecting a vehicle, $\hat{p}_{tei}$, on each day $t = 1, \dots, T = 365$ along each edge $e = 1, \dots, E = 286$ for each vehicle $i = 1, \dots, N_q \approx 135k$ in the population. This can be seen as a form of mass imputation or superpopulation modeling approach (Elliott & Valliant, 2017).

Simply summing the predicted probabilities per day and edge ($\hat{Y}_{te} = \sum_i \hat{p}_{tei}$) will yield a biased estimate. The term bias is used in this study to describe the difference between the predicted and actual traffic counts, which is a more loose definition than statistical bias meaning the difference between the expected value of an estimator and the true parameter. One apparent solution would be to predict the probability of detection only for the vehicles in the sample and weigh them with the inverse inclusion probability $\pi_{te} = \frac{2N_{te}^+}{N_q}$ (see Section 4.2). N_{te}^+ is, however, unknown for edges without a sensor. Averaging the inclusion probabilities across edges with a sensor ($\pi_t = \frac{2\sum_{e=1}^{K} N_{te}^+}{KN_q}$) unjustly assumes homogeneity over egdes and might do more harm than good. More fundamentally, weighting is not an option because along edges without a sensor there is no sample that can be weighted.

Instead, here we apply a—rather crude—bias correction through calibration. The predicted probabilities are ordered per day and edge, binned into $B = 10$ deciles, and summed per bin b. The result $\hat{Y}_{teb}$ is weighted with calibration weight w_{tb} before being summed over bins to yield the targeted estimate $\hat{Y}_{te}$.

The calibration weight w_{tb} is determined as follows. Each of the K models trained on a sample from $K - 1$ edges with a WiM sensor is used to predict $\hat{p}_{tki}$ along the excluded edge k, for each vehicle in the population. Per day and test set k, the predicted probabilities are ordered, binned into B deciles, and summed per bin. The result $\hat{Y}_{tkb} = \sum_{i \in b} \hat{p}_{tki}$ is compared with the observed count $Y_{tkb} = \sum_{i \in b} y_{tki}$ by the WiM sensor along edge k. For instance, if 100 vehicles are predicted but the WiM sensor detects only 80, the calibration weight w_{tkb} for that day, edge, and the bin will be 0.8. This procedure is repeated for each day, edge with WiM sensor and bin. The average calibration weight is obtained by summing numerator and denominator over K edges with a WiM sensor.

The estimation method is summarized in Equation 1:

$$\hat{Y}_{te} = \sum_{b} w_{tb} \hat{Y}_{teb} \tag{1}$$
$$\hat{Y}_{teb} = \sum_{i \in b} \hat{p}_{tei}$$
$$w_{tb} = \frac{\sum_k \sum_{i \in b} y_{tki}}{\sum_k \sum_{i \in b} \hat{p}_{tki}}$$

Figure 5 shows reliability plots of the uncorrected predicted counts $\hat{Y}_{tkb}$ plotted against the observed counts Y_{tkb} for one arbitrarily chosen day (2015-09-21), per scenario. Each gray line is a test set k, and each point along a line is a bin b, where the point $b = B$ is highlighted. The red line is the average across K test sets (the denominator against the numerator of w_{tb}). Summing over the bins would yield the actual and uncalibrated estimate of the daily count (cf. Fig. 3).

For all six scenarios, the number of vehicles is highly overestimated. When models are trained on the balanced sample, the overestimation is larger, but the variation between test sets is smaller. Probability partitioning does not prevent the overestimation. Most curves level off, which means that the overestimation is smaller for larger bins. These findings do not only apply to the one day shown but are found throughout the entire year.

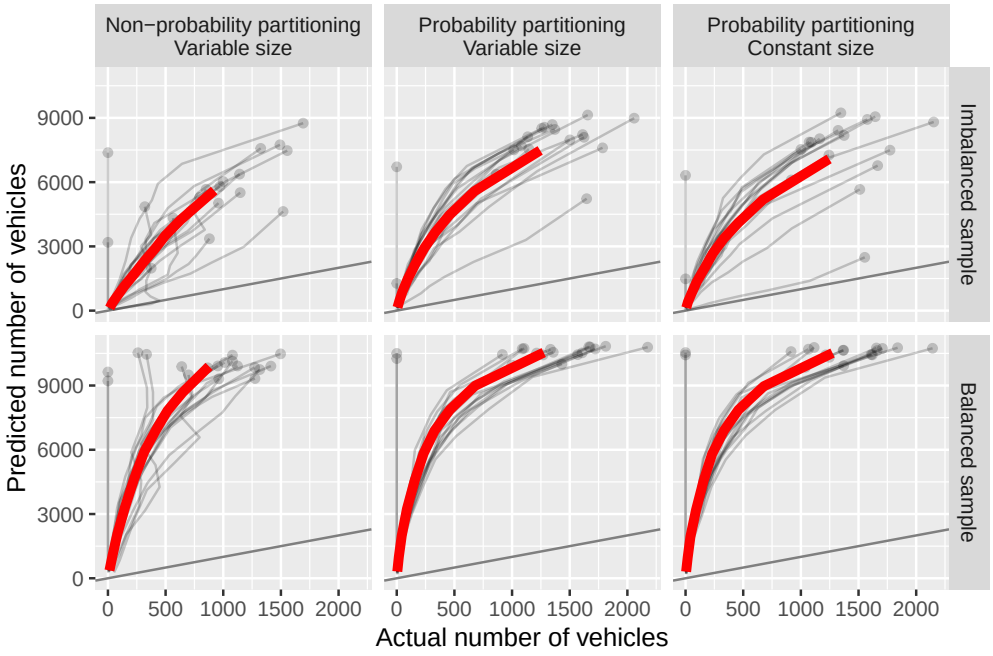


Figure 5: Reliability plots for six scenarios (panels), one day (2015-09-21) and $K = 18$ test sets (gray lines with point where $b = B$). Red line is the average across test sets. The diagonal $y = x$ indicates perfect predictions. The gray dots highlight $b = B$.

Figure 6 shows the calibration weights w_{tb}, i.e., the ratio between the observed and predicted count, per scenario, day, and bin. Since the calibration weights are always less than 1, there is overestimation throughout the entire year. The overestimation is larger (weights are further from 1) for models trained on the balanced sample, weekends, and lower bins. We reconsider this finding in the discussion, which is probably related to the applied sampling scheme.

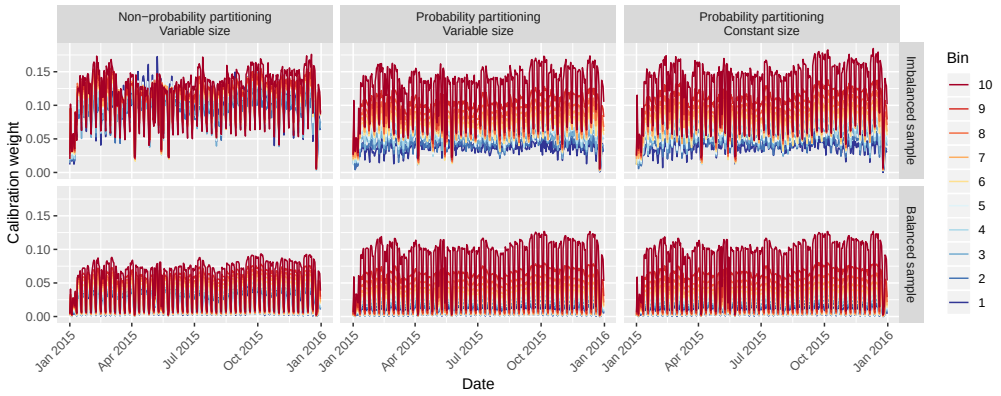


Figure 6: Daily calibration weights for the six considered scenarios using 10 bins.

5 Results

First, the overall achieved quality per scenario (Section 5.1) is shown, second, the feature importance is reported (Section 5.2), and third, the inferred network traffic is visualized (Section 5.3).

5.1 Overall quality

Figure 7 shows per scenario the quality of the model predictions across $K = 18$ test sets according to the metrics discussed in Section 4.3. A score of 0 corresponds to randomly guessing vehicle detection with a probability equal to the fraction of vehicles detected in the test set. Hence, the quality score can be interpreted as a fraction of the way from random guessing to perfect prediction. A negative score means that the model is performing worse than random guessing. Remember that here we evaluate the individual predictions of the sample, not the population estimates.

Using the imbalanced sample and non-probability partitioning (scenario 1, top left panel), the achieved quality was about halfway between random guessing and perfect prediction according to (min-max normalized) accuracy, area under the ROC curve, Youden's J, and negative F1. However, according to markedness, MCC, and positive F1, the model performed only marginally better than random guessing. There was substantial variation in performance across test sets.

Using the balanced sample (scenario 2, bottom left panel), the score increased mainly on those metrics indicating poor performance in scenario 1. Hence, those metrics (markedness, MCC, and F1 of the positive—i.e., rare—class) tell best if the model can outperform random guessing in case of class imbalance. In both scenarios, two or three test sets show that models can seriously underperform when trained on a non-probability sample of the data.

Using probability partitioning (scenarios 3 and 4, middle panels), the most striking effect is that model performance became consistent across test sets. Of course, this is to be expected since the test sets are now well represented in the training sets. Median performance, however, hardly increased. Markedness remained relatively unreliable when trained on an imbalanced sample. Removing the variation in test set size (scenarios 5 and 6, right panels) did not affect model performance.

To sum up, when the sample is balanced, and the test data are well represented in the training data, models performed about halfway between random guessing and perfect prediction. Class imbalance seriously lowered model performance, but not all metrics pick up this signal. Training on a non-probability sample also increased the risk of poor performance.

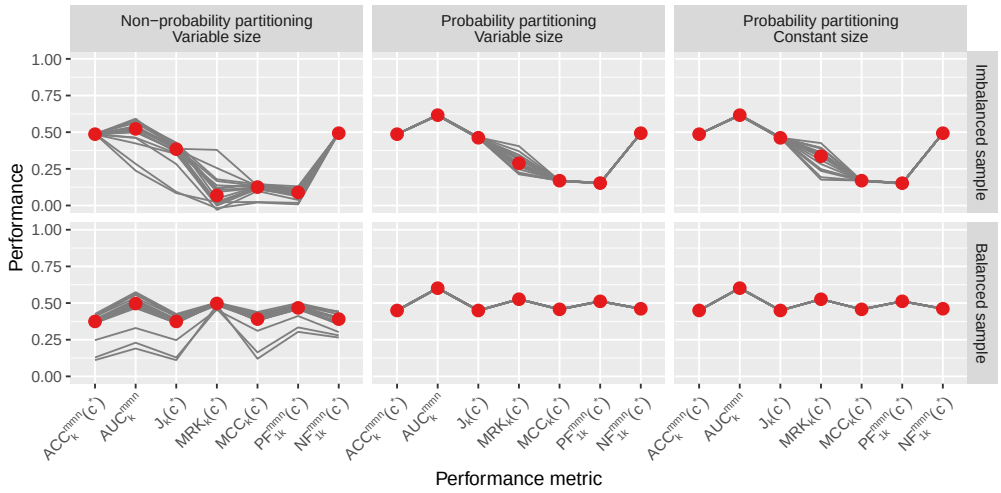


Figure 7: Model performance for six scenarios (panels) and $K = 18$ test sets (gray lines) according to seven performance metrics. Red dot is the median across test sets.

5.2 Importance of features

In this section, the importance of features per scenario is shown (Figure 8; see Appendix A for feature descriptions). Feature importance is quantified by a feature's relative gain in accuracy—i.e., the reduction in the negative log-likelihood—across splits and trees. The gains add up to 1 across features per scenario and test set. Only the top-20 out of the 117 features is shown, representing all groups of features (time, graph, traffic intensity, vehicle, and owner) and accounting for about 75% of the total gain. The vertical axis shows the features with the order aligned to the importance in scenario 1 (top left panel) for comparison between scenarios.

The two most important features are the maximum mass of the trailer and vehicle age. The top-20 contains mostly vehicles features (6) and owner features (8), only three graph features (edge weight and strength and closeness of an edge's origin vertex), two traffic intensity features (kurtosis and number of loop sensors), and one time feature (weekend indicator). Weather features, mean traffic intensity, and the graph features of an edge's destination vertex are of limited importance.

It is apparent that the weekend indicator is important in all models trained on an imbalanced sample (scenarios 1, 3, 5; top panels) but is of no importance in models trained on a balanced sample (scenarios 2, 4, 6; bottom panels). We will reconsider this finding in the discussion, as it can probably be explained by the sampling scheme used.

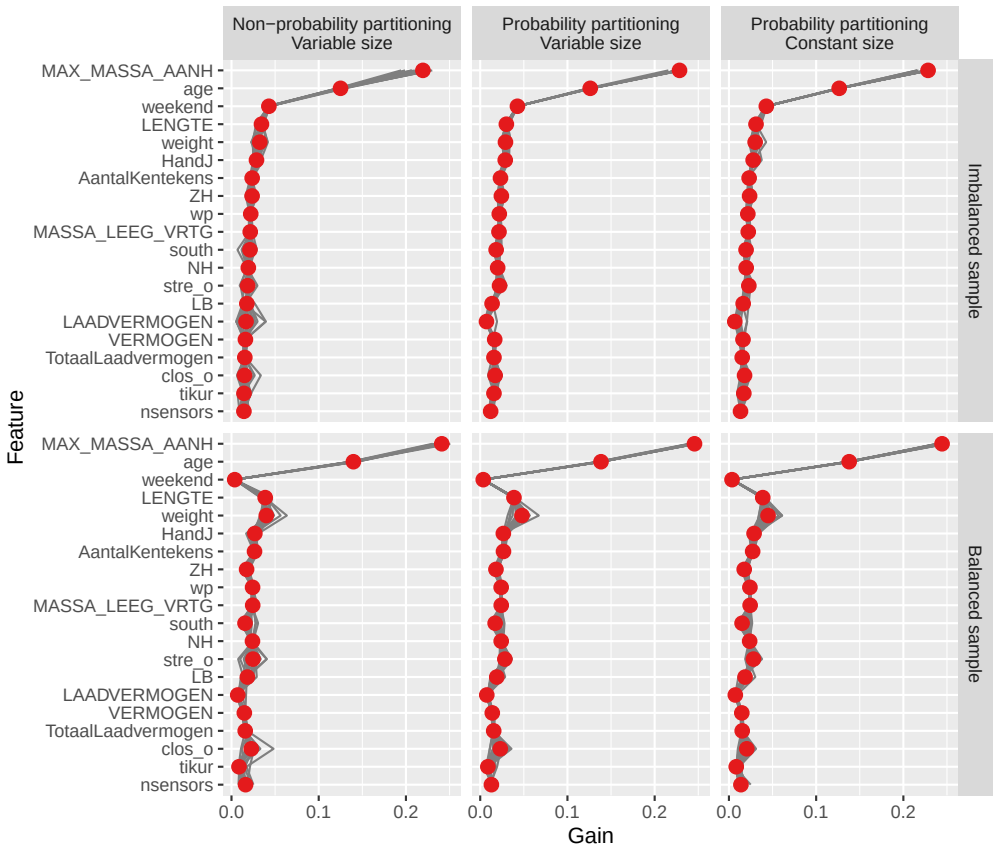


Figure 8: Feature importance for six scenarios (panels) and $K = 18$ test sets (gray lines). Red dot is the median across test sets.

5.3 Inferred network traffic

Figure 9 shows the inferred daily Dutch freight traffic across the Dutch road transport network ($\hat{Y}_{te}$ in Eq. 1) throughout 2015. The control panel allows playing the full movie, to go step-wise through the movie, or to stop the movie. On top of the figure, the day and date are shown. Low counts are shown in dark blue colors, while increasing counts start with turquoise, going over in green, yellow, up to orange, and red for the highest counts. Since predictions are made at the micro-level, similar movies can be constructed for any feature used, for instance type of truck or truck owner.

As the movie shows, estimated traffic is low during weekends and public holidays (e.g., Easter Monday 2015-04-06). This is mostly attributable to the calibration (see Fig. 6). Most traffic is concentrated in the western and southern regions, which might be expected given the location of the port of Rotterdam, Antwerp, and the Ruhr area. The model also predicts more traffic in the eastern and northern direction than the other way around, which also corroborates the notion that freight enters the port by sea and is distributed by land.

The inferred network traffic should, however, be interpreted with caution, given the poor quality of the predictions in the sample test sets (Fig. 7) and strong calibration weights (Fig. 6).

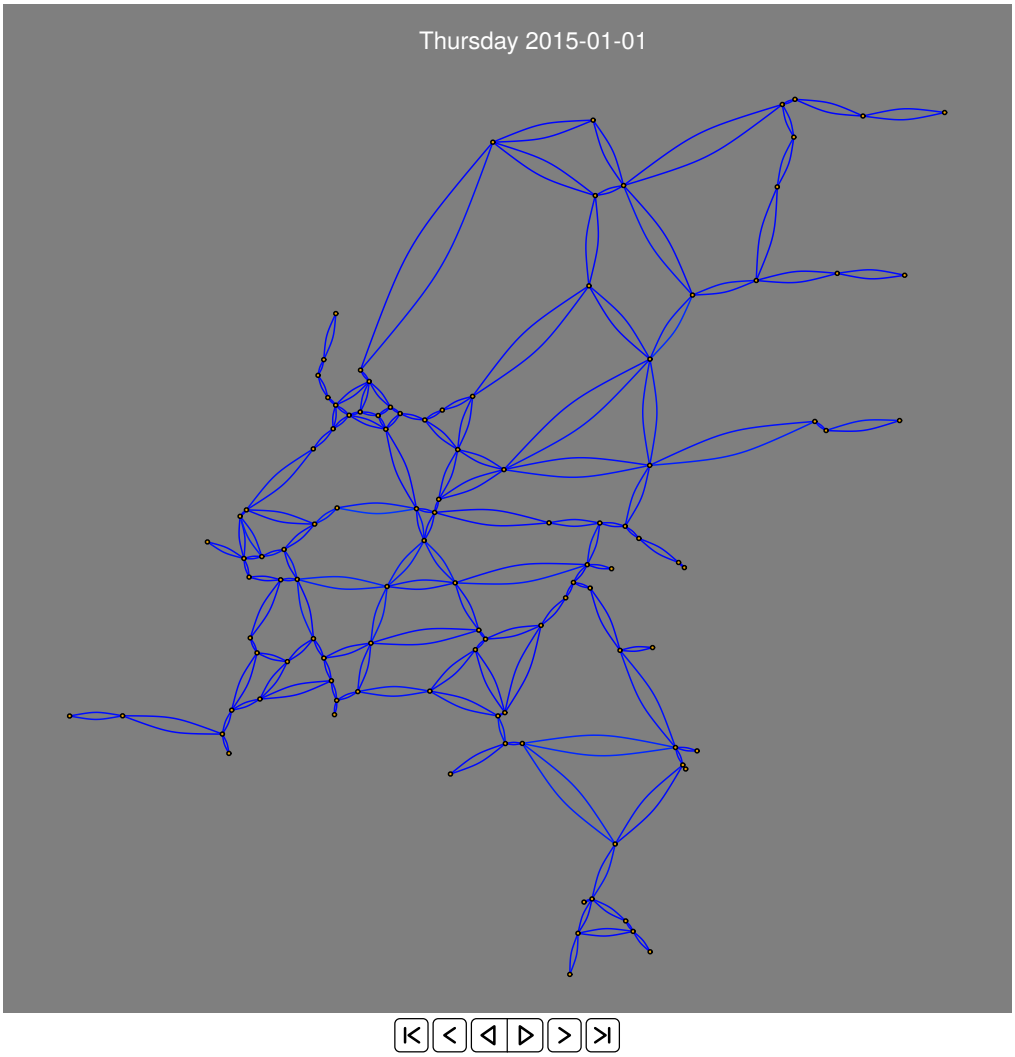


Figure 9: Movie of network traffic inferred from the imbalanced non-probability sample.

6 Discussion

General

This study aimed to produce official statistics using big data from road sensors as the primary source for the target variable. More specifically, the study aimed to infer network freight traffic from sensors located on Dutch freeways without a sampling design. Without a sampling design, design-based approaches cannot be deployed. Instead, the missing data was imputed using an algorithmic inference approach. The relationship between features and vehicle detection was learned and extrapolated to edges without sensors. The result was demonstrated in Section 5.3 by a movie of the inferred network traffic throughout 2015.

The method yields estimates of traffic across the network without the costs, response burden, and processing time of a diary survey. However, the quality of the estimates may not meet the quality standards in official statistics. Our simulation has revealed four challenges: non-probability sampling, class imbalance, quality of the data, and the modeling approach. Additionally, we will briefly discuss model explainability.

Non-probability sampling

This study's initial challenge—and that of other big data sources in general—is that the data is a non-probability sample of the statistical population. Although there was plenty of data (about 14 million

recorded Dutch transport vehicles), vehicles were only recorded along a non-random sample of 18 edges in the network. The non-probability nature of the data caused substantial variation in model performance across test sets. The big number of records, the availability of license plate as linkage key, and our intense effort to collect over 100 potential predictors could not prevent this. It confirms the well-known risk of extrapolating a model to domains that are not well represented in the data (Kitchin, 2015). In addition, the non-probability nature of the data might also bias the quality metrics, because the observations are not independent and identically distributed.

Class imbalance

The initial dataset is severely imbalanced, i.e., only about 2% of the vehicles in the register is detected by the WiM sensors. By balancing the sample and relating the model performance to frequency-dependent random guessing, our simulations showed that the class imbalance seriously compromised model performance. Matthews' correlation coefficient (MCC) and the min-max normalized harmonic mean of recall and precision of the rare class (PF_1^{mmn}) proved the most insightful performance metrics. Although balancing the sample can improve model performance on test sets of the balanced sample, the calibration weights show that balancing the sample impairs model performance when making inference from the sample to the population.

The big dataset (over 800 mln records), combined with the severe class imbalance and suspicion about the quality of low counts, made us decide to sample with an inclusion probability depending on the target variable. Busy days and edges were therefore overrepresented in both the balanced and imbalanced sample. The samples allowed us to study the effects of non-probability partitioning and class imbalance in the simulation, but a rather crude bias correction was needed to make inference to edges without a sensor. Ideally, inclusion probabilities can be based on strong predictors, which allows the weighting of the sample predictions. The available features, however, did not qualify.

Data

Even if the models are trained on a probability sample without class imbalance, model performance did not exceed halfway between random guessing and perfect prediction. The quality of the input data needs to be improved to improve model predictions. First, the irregular time series of the recorded counts suggest that low counts could actually be measurement errors. Input data quality would benefit from a way to discern low traffic intensity from sensor malfunctioning. Second, more WiM sensors would obviously alleviate the missing data problem. Installing more systems would involve considerable costs. Privacy-preserving record linkage could be a way to incorporate data from WiM sensors already installed in Belgium and Germany. Third, stronger predictors would allow for better mapping (e.g., adding month indicators). For instance, Google mobility reports, cross-border information, and the province of an edge's origin and destination vertex. A minor improvement would be to replace the haversine distance between vertices with the actual driving distance.

Modeling approach

Model predictions could also improve by improving the modeling approach. First, the current model could be improved by a more thorough optimization of the hyperparameters. Second, possible alternative methods like graph neural networks might be better able to deal with the spatiotemporal dependencies. Third, the inference problem could be simplified by modeling an aggregate level such as the total number of vehicles per day per edge. It could also be made more complex by microsimulation. Finally, if all else fails, the idea of big data as a primary source should be abandoned. Combining big data with survey data opens up a wider array of methods (Elliott & Valliant, 2017; Lohr & Raghunathan, 2017; Vaillant, 2020), at the expense of survey costs, burden and time.

Explainability

Model explainability is an important aspect in machine learning, where regression coefficients do not capture the relationship between feature and prediction. In this study, explainability was limited to each feature's relative contribution to reducing prediction error. Further analysis would be required to understand better what happens inside the black box producing individual probabilities from feature values. For example, partial dependence plots, SHAP values and counterfactual explanations (Molnar, 2021).

7 Conclusion

This study has applied a superpopulation modeling approach to potentially use big data as the primary source for producing official statistics. However, the non-probability nature of the data and class imbalance cause severe challenges. Despite 27 million records, more than 100 features, and a modern machine learning algorithm, the inferred network traffic may not meet the quality standards for official statistics. Making inference from a non-probability sample increases the risk of producing biased estimates. The negative class imbalance makes it difficult for models to defeat random guessing.

Turning lead into gold is possible (literally by removing three protons, three electrons and four to eight neutrons depending on the lead isotope), but sometimes the costs outweigh the benefits. Producing official statistics from sensor data as the primary source would be helped by a sampling design or features that can explain the data generating mechanism, and by classification models that can defeat random guessing when predicting a rare event.

References

Baker, R. et al. (2013). "Summary report of the AAPOR task force on non-probability sampling". In: *Journal of Survey Statistics and Methodology* 1.2, pages 90–105.

Breiman, L. (1997). *Arcing the Edge*. Technical Report 486. Statistics Department, University of California, Berkeley.

Breiman, L. (2001). "Statistical modeling: The two cultures". In: *Statistical Science* 16.3, pages 199–231.

Buelens, B., J. Burger, and J.A. van den Brakel (2018). "Comparing inference methods for non-probability samples". In: *International Statistical Review* 86.2, pages 322–343.

Buelens, B. et al. (2012). "Shifting paradigms in official statistics: From design-based to model-based to algorithmic Inference". In: *Statistics Netherlands (CBS) Discussion Paper, The Hague/Heerlen* 201218.

Buelens, B. et al. (2014). "Selectivity of Big Data". In: *Statistics Netherlands (CBS) Discussion Paper, The Hague/Heerlen* 2014–11.

Buiten, G., E. de Jonge, and F. P. Pijpers (2018). "Toepassingsmogelijkheden van complexiteitstheorie bij het CBS". In: *Statistics Netherlands (CBS) Discussion Paper, The Hague/Heerlen*.

Burger, J. and Q. Meertens (2020). "The algorithm versus the chimps: On the minima of classifier performance metrics". In: *Proceedings of BNAIC/BeneLearn 2020*. Edited by L. Cao, W. Kosters, and J. Lijffijt. Leiden University, pages 38–55.

Centraal Bureau voor de Statistiek (2017). *Basisbestanden Goederenwegvervoer 2015*. CBS publicaties, The Hague/Heerlen.

Chawla, N.V. et al. (2002). "SMOTE: Synthetic minority over-sampling technique". In: *Journal of Artificial Intelligence Research* 16, pages 321–357.

Chawla, N.V. et al. (2003). "SMOTEBoost: Improving prediction of the minority class in boosting". In: *Knowledge Discovery in Databases: PKDD 2003*. Berlin: Springer, pages 107–119.

Chen, T. and C. Guestrin (2016). "XGBoost: A scalable tree boosting system". In: *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, California, USA: Association for Computing Machinery. DOI: 10.1145/2939672.2939785.

Cornesse, C. et al. (2020). "A Review of conceptual approaches and empirical evidence on probability and nonprobability sample survey research". In: *Journal of Survey Statistics and Methodology* 8.1, pages 4–36. DOI: 10.1093/jssam/smz041.

Csardi, G. and T. Nepusz (2006). "The igraph software package for complex network research". In: *InterJournal* Complex Systems, page 1695. URL: http://igraph.org.

Daas, P.J.H. et al. (2015). "Big data as a source for official statistics". In: *Journal of Official Statistics* 31.2, pages 249–262.

De Broe, S. et al. (2020). "Updating the paradigm of official statistics: New quality criteria for integrating new data and methods in official statistics". In: *Statistical Journal of the IAOS* Pre-press.Pre-press, pages 1–18.

Elliott, M.R. and R. Valliant (2017). "Inference for nonprobability samples". In: *Statistical Science* 32.2, pages 249–264.

Federal Highway Administration (2007). *Effective Use of Weigh-in-Motion Data: The Netherlands Case Study*.

Friedman, J.H. (1999a). *Greedy function approximation: A gradient boosting machine*. Sequoia Hall, Stanford University, Stanford, CA.

Friedman, J.H. (1999b). *Stochastic gradient boosting*. CSIURO CMIS, Locked Bag 17, North Ryde NSW 1670.

Gootzen, Y.A.P.M., M.R. Roos, and B.O. Mussman (2020). "Combining data sources to gain new insights in mobility: A case study". In: Statistics Netherlands – Center for Big Data Statistics.Working paper 03–2020.

Hackl, P. (2016). "Big data: What can official statistics expect?" In: *Statistical Journal of the IAOS* 32.1, pages 43–52.

Harford, T. (2014). "Big data: A big mistake?" In: *Significance* 11.5, pages 14–19.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2nd edition. Springer-Verlag.

Hsu, L.-H. and C.-K. Lin (2009). *Graph Theory and Interconnection Networks*. Boca Raton: CRC Press.

Jungnickel, D. (2005). *Graphs, Networks and Algorithms*. 2nd edition. Berlin: Springer.

Kim, G.-H., S. Trimi, and J.-H. Chung (2014). "Big-data applications in the government sector". In: *Communications of the ACM* 57.3, pages 78–85.

Kitchin, Rob (2015). "The opportunities, challenges and risks of big data for official statistics". In: *Statistical Journal of the IAOS* 31.3, pages 471–481.

Klingwort, J. (2020). *Correcting Survey Measurement Error With Big Data from Road Sensors Through Capture-Recapture*. Dissertation, University of Duisburg-Essen. DOI: 10.17185/duepublico/72081.

Klingwort, J., B. Buelens, and R. Schnell (2019a). "Capture-recapture techniques for transport survey estimate adjustment using road sensor data". In: *Social Science Computer Review* Online first. DOI: doi.org/10.1177/0894439319874684.

Klingwort, J. et al. (2019b). "Graph-based inference from non-probability road sensor data". In: *Book of Abstracts of the 8th International Conference on Complex Networks and their Applications*. Edited by H. Cheri et al. Lisbon: International Conference on Complex Networks & Their Applications, pages 599–601.

Lohr, S.L. and T.E. Raghunathan (2017). "Combining survey data with other data sources". In: *Statistical Science* 32.2, pages 293–312.

Luque, A. et al. (2019). "The impact of class imbalance in classification performance metrics based on the binary confusion matrix". In: *Pattern Recognition* 91, pages 216–231. DOI: 10.1016/j.patcog.2019.02.023.

Molnar, C. (2021). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. https://christophm.github.io/interpretable-ml-book/.

National Data Warehouse for Traffic Information (2016). *NDW: A nationwide portal for traffic information. Retrieved from: https://www.ndw.nu/documenten/nl/*.

Powers, D.M.W. (2011). "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation". In: *Journal of Machine Learning Technologies* 2.1, pages 37–63.

Puts, M.J.H. et al. (2019). "Using huge amounts of road sensor data for official statistics". In: *AIMS Mathematics* 4.1, page 12. DOI: 10.3934/Math.2019.1.12.

Schnell, R. (2019). "Big Data aus wissenschaftssoziologischer Sicht: Warum es kaum sozialwissenschaftliche Studien ohne Befragungen gibt". In: *Erklärende Soziologie und soziale Praxis*. Edited by D. Baron, O.A. Becker, and D. Lois. Wiesbaden: Springer VS, pages 101–125.

Schreutelkamp, F.H. and G.L. Strang van Hees (2001). "Benaderingsformules voor de transformatie tussen RD- en WGS84-coördinaten". In: *Geodesia*, pages 64–69.

Shlomo, N. and H. Goldstein (2015). "Editorial: Big data in social research". In: *Journal of the Royal Statistical Society. Series A* 178.4, pages 787–790.

Tam, S.-M. and F. Clarke (2015). "Big data, official statistics and some initiatives by the Australian Bureau of Statistics". In: *International Statistical Review* 83.3, pages 436–448.

Vaillant, R. (2020). "Comparing alternatives for estimation from nonprobability samples". In: *Journal for Survey Statistics and Methodology* 8, pages 231–263.

van der Laan, J. (2019). *Measuring segregation using a social network of the Netherlands*. ODISSEI Community Conference 22 October 2019.

van Steen, M. (2010). *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen.

Wickham, H. (2019). *rvest: Easily harvest (scrape) web pages*. R package version 0.3.5. URL: https://CRAN.R-project.org/package=rvest.

Appendix

A Features

Table 4: Time features

Feature	Description	Unit or integer labels
Day		
$weekend_t$	indicator for weekend	
mon_t	indicator for Monday	
tue_t	indicator for Tuesday	
wed_t	indicator for Wednesday	
thu_t	indicator for Thursday	
fri_t	indicator for Friday	
sat_t	indicator for Saturday	
sun_t	indicator for Sunday	
Wind		
$DDVEC_t$	vector mean wind direction	degrees
$FHVEC_t$	vector mean wind speed	0.1 m/s
FG_t	24-h mean wind speed	0.1 m/s
FHX_t	highest hourly mean wind speed	0.1 m/s
FHN_t	lowest hourly mean wind speed	0.1 m/s
FXX_t	highest wind gust	0.1 m/s
Temperature		
TG_t	24-h mean temperature	0.1 °C
TN_t	minimum temperature	0.1 °C
TX_t	maximum temperature	0.1 °C
$T10N_t$	minimum temperature at 10 cm height	0.1 °C
Sunshine		
SQ_t	sunshine duration	0.1 hour
SP_t	sunshine duration relative to the longest possible sunshine duration	%
Q_t	global radiation	J / cm^2
Precipitation		
DR_t	duration of the precipitation	0.1 hour
RH_t	24-h sum of the precipitation	0.1 mm
RHX_t	highest hour sum of the precipitation	0.1 mm
Air pressure		
PG_t	24-h mean air pressure traced back to sea level	0.1 hPa
PX_t	highest hourly value of the air pressure traced back to sea level	0.1 hPa
PN_t	lowest hourly value of the air pressure traced back to sea level	0.1 hPa
Visibility		
VVN_t	minimum visibility	0: < 100 m 1: 100–200 m ... 50: 5–6 km 56: 6–7 km ... 80: 30–35 km ... 89: > 70 km
VVX_t	maximum visibility	same as VVN_t
Cloud cover		
NG_t	24-h mean cloud cover in eights (9: sky obstructed from view)	
Humidity		
UG_t	24-h mean relative humidity	%
UX_t	maximal relative humidity	%
UN_t	minimal relative humidity	%
Evapotranspiration		
$EV24_t$	Makkink reference crop evapotranspiration	0.1 mm

Table 5: Edge features

Feature	Description	Unit or integer labels
Network centrality		
$weight_e$	weight (inverse haversine or great-circle distance)	km^{-1}
$degr_o$	degree (number of outgoing edges) of origin vertex o	
$degr_d$	degree of destination vertex d	
$stre_o$	strength (total weight of outgoing edges) of origin vertex o	km
$stre_d$	strength of destination vertex d	km
$betw_o$	betweenness (number of shortest paths passing through) of origin vertex o	
$betw_d$	betweenness of destination vertex d	
$clos_o$	closeness (inverse of the average length of the shortest paths to all the other vertices) of origin vertex o	km^{-1}
$clos_d$	closeness of destination vertex d	km^{-1}
$vuln_o$	vulnerability (loss in efficiency—average inverse distance—when excluded) of origin vertex o	
$vuln_d$	vulnerability of destination vertex d	
$clus_o$	clustering coefficient (probability that adjacent vertices are connected) of origin vertex o	
$clus_d$	clustering coefficient of destination vertex d	
$page_o$	PageRank (probability that a randomly driving vehicle reaches the vertex when the likelihood of starting again at a random vertex is 15 %) of origin vertex o	
$page_d$	PageRank of destination vertex d	
Daily traffic intensity		
$timean_{te}$	mean (unstandardized first moment about zero) traffic intensity on day t measured by loop sensors mapped to edge e	day^{-1}
$tivar_{te}$	variance (unstandardized second moment about the mean) of traffic intensity on day t measured by loop sensors mapped to edge e	day^{-2}
$tiskew_{te}$	skewness (standardized third moment about the mean) of traffic intensity on day t measured by loop sensors mapped to edge e	
$tikur_{te}$	kurtosis (standardized fourth moment about the mean) of traffic intensity on day t measured by loop sensors mapped to edge e	
$timean_{t-1,e}$	mean traffic intensity on day $t-1$ measured by loop sensors mapped to edge e	day^{-1}
$timean_{t-7,e}$	mean traffic intensity on day $t-7$ measured by loop sensors mapped to edge e	day^{-1}
$timean_{ta}$	mean of $timean_{te}$ across all incoming edges of origin vertex o	day^{-1}
$timean_{tp}$	mean of $timean_{te}$ across all outgoing edges of destination vertex d	day^{-1}
$nsensors_e$	number of loop sensors mapped to edge e	

Table 6: Vehicle and owner features

Feature	Description	Unit or integer labels
Vehicle		
$AANT\_WIELEN_i$	number of wheels	
$diesel_i$	indicator for diesel	
$AANT\_CYL_i$	number of cylinders	
$VERMOGEN_i$	power	kW
$MAX\_MASSA\_VRTG_i$	maximum mass	kg
$MASSA\_LEEG\_VRTG_i$	mass when empty	kg
$MAX\_MASSA\_AANH_i$	maximum mass of trailer	kg
$max\_massa\_aanh0_i$	indicator for $MAX\_MASSA\_AANH_i = 0$ (unable to pull a trailer)	
$LAADVERMOGEN_i$	loading capacity	kg
$BREEDTE_i$	width	m
$LENGTE_i$	length	m
$lengte0$	indicator for $LENGTE_i = 0$ (missing)	
age_i	age	
$LEASE\_IND_i$	indicator for lease	
$juridical_i$	indicator for juridical entity	
$fueltruck_i$	indicator for fuel truck	
$boxtruck_i$	indicator for box truck	
$opentruck_i$	indicator for open truck	
$cattletruck_i$	indicator for cattle truck	
$refrigtruck_i$	indicator for refrigerator truck	
$othertruck_i$	indicator for other truck	
$specialveh_i$	indicator for (non-truck) special vehicle	
$tractor_i$	indicator for (non-truck) tractor	
Vehicle owner		
$BthruE_i$	indicator for active in energy or industry excl. construction	
$GthruI_i$	indicator for active in trade, transport or catering	
$MthruN_i$	indicator for active in corporate services	
$OthruQ_i$	indicator for active in government or healthcare	
$RthruU_i$	indicator for active in culture, recreation or other services	
A_i	indicator for active in agriculture, forestry or fishing	
$BthruF_i$	indicator for active in energy or industry	
$GandI_i$	indicator for active in trade or catering	
$HandJ_i$	indicator for active in transport or information	
$GthruN_i$	indicator for active in commercial services	
$OthruU_i$	indicator for active in non-commercial services	
$E3811_i$	indicator for active in collection of non-hazardous waste	
$G45112_i$	indicator for active in sale of cars or light motor vehicles, excl. import	
$H4941_i$	indicator for active in freight transport by road	
$H52291_i$	indicator for active as forwarder, shipbroker or charterer	
wp_i	number of working persons	0: 0 1: 1 2: 2 3: 3–4 5: 5–9 10: 10–19 20: 20–49 50: 50–99 100: 100–149 150: 150–199 200: 200–249 250: 250–499 500: 500-999 1000: 1000–1999 2000: 2000+
$AantalKentekens_i$	number of license plates	
$TotaalLaadvermogen_i$	total loading capacity	kg
$commercial_i$	indicator for commercial transport	
GR_i	indicator for located in the province of Groningen	
FR_i	indicator for located in the province of Friesland	
DR_i	indicator for located in the province of Drenthe	
OV_i	indicator for located in the province of Overijssel	
FL_i	indicator for located in the province of Flevoland	
GL_i	indicator for located in the province of Gelderland	
UT_i	indicator for located in the province of Utrecht	
NH_i	indicator for located in the province of Noord-Holland	
ZH_i	indicator for located in the province of Zuid-Holland	
ZL_i	indicator for located in the province of Zeeland	
NB_i	indicator for located in the province of Noord-Brabant	
LB_i	indicator for located in the province of Limburg	
$north_i$	indicator for located in the north (GR, FR, DR)	
$east_i$	indicator for located in the east (OV, FL, GL)	
$west_i$	indicator for located in the west (UT, NH, ZH, ZL)	
$south_i$	indicator for located in the south (NB, LB)	
$missingowner_i$	indicator for missing owner	

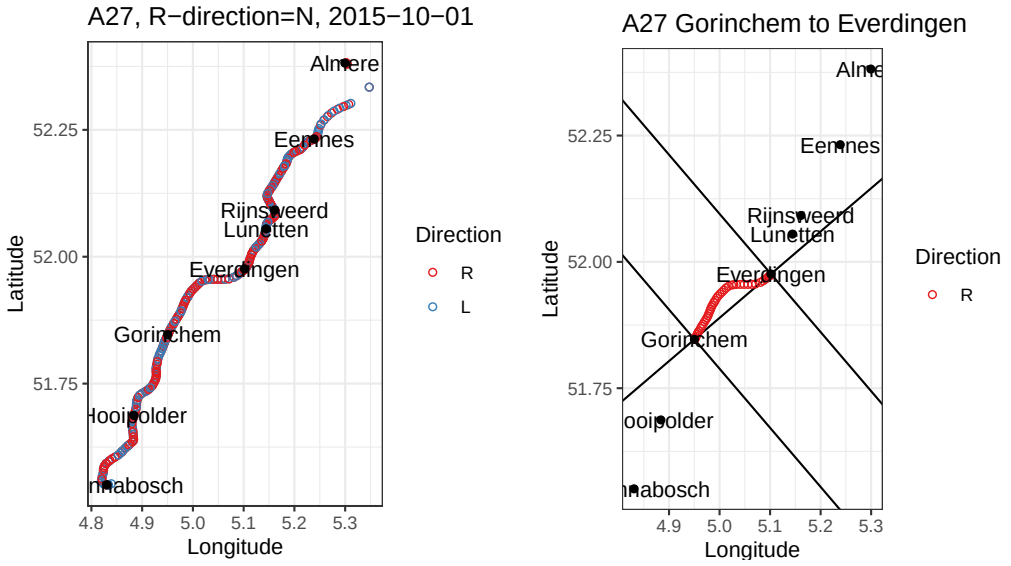
B Assigning loop sensors to edges

The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is directed so traffic intensity from vertex o to vertex d may differ from traffic intensity from vertex d to vertex o. The traffic loop database contains for direction 'R' a compass direction per road number, using the 8-wind compass rose (N/NE/etc.). The edge list contains per road number two vertices A and Z that have only one incoming and one outgoing edge (except ring A10, which is treated separately). If the compass direction from A to Z agrees with the compass direction in the traffic loop database, sensors with direction 'R' are mapped from A to Z (and sensors with direction 'L' from Z to A). If not, sensors with direction 'R' are mapped from Z to A (and sensors with direction 'L' from A to Z) (Fig. 10a).

For ring A10 the traffic loop database contains for direction 'R' compass direction 'CW' (clockwise). After manual inspection of the map, sensors with direction 'R' were therefore mapped to edges Coenplein–Watergraafsmeer–Amstel–De Nieuwe Meer–Coenplein (and sensors with direction 'L' to edges Coenplein–De Nieuwe Meer–Amstel–Watergraafsmeer–Coenplein).

Vertices of the graph have GPS coordinates (longitude and latitude from www.coordinatenbepalen.nl) in the WGS84 system. The sensors in the traffic loop database, however, have RD coordinates according to the EPSG:28992 system, where base point Amersfoort has coordinates 155 km East and 463 km North. RD coordinates were transformed to the ellipsoidal WGS84 coordinates using the approximation by Schreutelkamp and Strang van Hees (2001). Coordinates of point p are denoted (x_p, y_p).

To map loop sensors to edge $\overrightarrow{od}$, we first draw a straight line between vertices o and d (Fig. 10b): $y = a + bx$, where slope $b = \frac{y_d - y_o}{x_d - x_o}$ and intercept $a = y_o - bx_o = y_d - bx_d$. Next, three lines are drawn perpendicular to the od-line: one through vertex o: $y = a_o - \frac{1}{b}x$, where $a_o = y_o + \frac{1}{b}x_o$, one through vertex d: $y = a_d - \frac{1}{b}x$, where $a_d = y_d + \frac{1}{b}x_d$ and one through loop sensor s: $y = a_s - \frac{1}{b}x$, where $a_s = y_s + \frac{1}{b}x_s$. Sensor s is assigned to edge $\overrightarrow{od}$ if $min(a_o, a_d) < a_s < max(a_o, a_d)$.



(a) Loop sensors along state road A27

(b) Loop sensors mapped to edge Gorinchem–Everdingen

Figure 10: Example of assigning loop sensors to edges of the graph.

If a loop sensor is mapped to more than one edge, it is mapped to the nearest edge. For that we introduce intersection t, where $y_t = a + bx_t = a_s - \frac{1}{b}x_t$, so $x_t = \frac{a_s - a}{b + \frac{1}{b}}$. The distance from loop sensor s to intersection t is $||\overrightarrow{st}|| = \sqrt{||\overrightarrow{os}||^2 - ||\overrightarrow{ot}||^2}$, where $||\overrightarrow{os}|| = \sqrt{(x_s - x_o)^2 + (y_s - y_o)^2}$ and $||\overrightarrow{ot}|| = \sqrt{(x_t - x_o)^2 + (y_t - y_o)^2}$.

Colophon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague
www.cbs.nl

Prepress
Statistics Netherlands

Design
Edenspiekermann

Enquiries
Telephone: +31 88 570 70 70
Via contact form: www.cbs.nl/infoservice