VŠB – Technical University of Ostrava
Faculty of Electrical Engineering and Computer Science
Department of Applied Mathematics

Modelling of traffic flow using modern stochastic approaches

Modelování dopravního toku s využitím moderních stochastických nástrojů

2017 Ing. Lukáš Rapant

I would like to express my gratitude to my tutor prof. Ing. Radim Briš, CSc. for his guidance and patience and to my wife and parents for their support.

Abstrakt

Spolu s rostoucím objemem provozu narůstá význam předpovědi stavu dopravního provozu. Schopnost předvídat dopravní rychlost a hustotu v krátkém až střednědobém horizontu je jednou z hlavních úkolů každého systému pro řízení dopravy. Tato predikce může být použita k řízení provozu, a to jak k prevenci vzniku dopravních zácp, tak k minimalizaci jejich dopadu. Tyto informace jsou také užitečné pro plánování jízdních tras. Předpověď stavu dopravního provozu není snadným úkolem, protože dopravní tok je velmi obtížné popsat numerickými rovnicemi. Dalším možným přístupem k předpovědi provozního stavu je použití historických údajů o provozu a jejich propojení s aktuálním stavem pomocí vhodného statistického přístupu. Tento úkol však komplikuje složitá povaha dopravních dat, která mohou být z různých důvodů poměrně nepřesná. Tato práce je zaměřena na nalezení algoritmů, které mohou využívat cenné informace obsažené v dopravních údajích z dálnic ČR za účelem vytvoření krátkodobých předpovědí rychlosti provozu. Mnou navrhované algoritmy jsou založeny na moderních stochastických přístupech jako jsou skryté Markovovy modely, dynamické bayesovské sítě a Kalmanovy filtry. Tyto modely dokáží přirozeně zachytit zákonitosti dopravního provozu a vzít v úvahu neznámou nejistotu zatěžující dopravní data.

Klíčová slova: Dopravní modelování, predikce časových řad, Bayesovské sítě, Kálmánovy filtry, skryté Markovské modely, Monte Carlo simulace, Markovské řetězce

Abstract

The importance of traffic state prediction steadily increases together with growing volume of traffic. The ability to predict traffic speed and density in short to medium horizon is one of the main tasks of every Intelligent Transportation System. This prediction can be used to manage the traffic both to prevent the traffic congestions and to minimize their impact. This information is also useful for route planning. Traffic state prediction is not an easy task given that the traffic flow is very difficult to describe by numerical equations. Other possible approach to traffic state prediction is to use historical data about the traffic and relate them to the current state by application of some form of statistical approach. This task is, however, complicated by complex nature of the traffic data, which can, due to various reasons, be quite inaccurate. This thesis is focused on finding the algorithms that can exploit valuable information contained in traffic data from Czech Republic highways to make a short-term traffic speed predictions. My proposed algorithms are based on modern stochastic approaches like hidden Markov models, dynamic Bayesian networks, ensemble Kalman filters, Monte Carlo simulation and Markov chains. These models are naturally able to capture all complexities in the traffic and incorporate uncertainty of the traffic data.

Key Words: Traffic modelling, time series prediction, Bayesian networks, Kalman filters, hidden Markov models, Monte Carlo simulation, Markov chains

Contents

List of abbreviations

BN	–	Bayesian network
DBN	–	Dynamic Bayesian network
KF	–	Kalman filter
EnKF	–	Ensemble Kalman filter
FCD	–	Floating car data
CPD	–	Conditional probability distribution
EM	–	Expectation-maximization
LWR	–	Lighthill, Whitham and Richards numerical model
TMC	–	Traffic Message Channel
GNSS	–	Global Navigation Satellite System
GPS	–	Global positioning system
HMM	–	Hidden Markov model
CTM-v	–	Cell transmission model - velocity
CPT	–	Conditional probability table
DE	–	Differential evolution
pdf	–	Probability distribution function
PDE	–	Partial differential equation
MAP	–	Maximum a posteriori probability
MLE	–	Maximum likelihood estimate
MAE	–	Mean absolute error
RMSE	–	Root mean square error
CFL	–	Courant-Friedrichs-Lewy

List of Figures

List of Tables

1 Introduction

One of the main tasks of Intelligent Transportation Systems is to predict state of the traffic in short to medium horizon. This prediction can be then applied in various areas of traffic modelling, analysis, and management, for example travel time prediction and traffic congestion prevention. During work on RODOS [1] project there has arisen a need for a traffic flow model that will be capable of macroscopic traffic modelling and short-term prediction. This traffic model must be able to model motorway traffic. It would also be desirable if it was able able to model traffic on the arterial roads in the big cities, but availability and quality of the data from the cities is still uncertain, so this work is focused on motorway traffic. The short-term traffic state prediction means ability to predict the traffic physical quantities in the near future (i.e. up to half an hour). These quantities are traffic speed, intensity and density (description of these quantities can be found in [18]). Ideal predication algorithm should be able to predict all these quantities; however, this ideal is very difficult or impossible to achieve. This model is also required to be able to run accurately on the available data. These data come from two general sources: stationary sensors and floating car data (FCD). The outline of the thesis is following. Goals of the thesis are set in the Section 2. Current and historical state of traffic state predictions is described in Section 3. Section 4 contains brief description of the available data sources. Section 5 presents basic approaches to traffic state predicting and proposes solutions to this problem. Section 6 describes theoretical foundations of these methods. Next, Section 7 deals with a problem of statistical analysis of the existing data sources with the aim on analysis of relation between both data sources. Section 8 presents developed algorithms and their experimental results. This thesis is concluded by discussion of these results and conclusion, which contains proposals for improvements and further research and development.

[1] http://www.rodos-it4i.cz/defaultEN.aspx

2 Thesis goals

The main aim of the thesis is to propose model or set of models, that can do a short-term traffic state predictions with as much detail as possible utilizing as much data as possible. This main aim of this thesis can be divided into following goals:

1. Verification that the data follow theoretical patterns and assessment of relationship between data sources. Statistical analysis will be utilized to verify that the data follow theoretical patterns which should hold true for any data regarding the traffic flow. Also, a method for comparison of measurements from different kind of sensors must be found and implemented in order to get as much information from the data as possible. The problem is that some of the sensors are working in a fixed place and measuring profile while the others are aggregating measurements over the specific part of the road. This configuration of data sources may or may not prove to be problematic and have to be thoroughly analysed. Results from this step will be utilized in the developed algorithms.

2. Finding, modification and implementation of models for short term traffic state prediction. These algorithms must be able to make some short-term traffic state predictions. They should also exploit as much of the available data sources mentioned in the introduction as possible because despite having multiple data sources, each of them has some drawbacks that can be overcome.

3. Validation of the models. All proposed models will be tested on historical data and their accuracy will be assessed. Their results will be compared and the better one would probably be used in the RODOS system and possibly in other projects of ADAS laboratory of IT4Innovations like ANTAREX[2]. The method for this comparison will be consulted with the traffic experts. If both models perform well, they may be used in parallel to provide assurance when one of them fails.

[2] http://www.antarex-project.eu/

3 State of the Art

A lot of work has already been done in the area of traffic state modelling and predictions. Chosen papers and researches relevant to my work divided into 3 parts. In the first part I present some articles about the traffic state modelling without use of the Bayesian approach. In the second part, the articles with Bayesian network approach to traffic modelling are presented. In the final part, the articles about traffic modelling with the Kalman filters are briefly mentioned.

Georgescu et. al. [19] propose to use a multi-level explanatory model (MTM) with linear regression to make a short-term traffic speed prediction. They use stationary sensors as their data source. On the other hand, Gopi et. al. [22] propose to use Bayesian Support Vector Regression (BSVR). BSVR has the advantage in measuring uncertainty of the prediction. They also work with stationary senor data. On the other hand, De Fabritiis et. al. [13] work with floating car data. They propose pattern matching approach in combination with artificial neural network and they proved this approach feasible, but requiring higher penetration of the traffic by floating cars. Paper written by Jiang et. al. [34] presents a comparison of several approaches to the problem of traffic speed prediction. They compare models coming from machine learning like artificial neural networks to the statistical models like ARIMA and VAR. They conclude that both kinds of models have several advantages and disadvantages, with machine learning methods coming slightly ahead. Data used for their predictions are coming from the stationary sensors. Asif et. al. [3] present yet another approach to the traffic speed prediction using ν-Support Vector Regression using clustering approach to historical data and prove it to be preferable to artificial neural network. Article [14] written by Doniec, Mandiau, Piechowiak and Espie as a typical representative of modern multi-agent approach has been chosen. They propose multi-agent model, which is able to deal realistically with complicated situations like road junctions by modelling various behavioural types of drivers. I am not going to mention more multi-agent approaches because they require different kind of data than I possess. Tang, Huang and Shang propose in their article [68] a variant of car-following traffic flow model which takes into consideration driver's forecast effect. The car-following models are microscopic models that are defined by ordinary differential equations describing the complete dynamics of the vehicles' positions and velocities. Similar article [51] was written by Peng, Cau, Liu and Cao with focus on numerical properties of the model. These models are again impractical for us because of the data. Article written by Wong and Wong is an example of fluid dynamics traffic flow modelling. They are using Lighthill, Whitham and Richards numerical model (LWR) with heterogeneous drivers. They have proven that this form of model better simulates a lot of phenomena like different speeds of the lanes. This model demands data, that I do not have but the special version of this model will be interesting for us and is described later. Yet, another example of non-Bayesian traffic modelling is article written by Yan et. al. [81] who used Neural Network to make short term traffic flow predictions.

A lot of works deal with the problem of traffic state modelling by application of some form of

Bayesian network or some of its subclasses. Wang and Kuang [74] proposed a traffic prediction method based on complex event processing and Adaptive Bayesian networks. They use synthetic data from the SUMO traffic simulator to evaluate their algorithms. On the other hand, Hoong et al. [29] propose to use Bayesian Network for road condition prediction. They presented a set of evidences that could potentially be utilized for road condition prediction and constructed a Bayesian Network model to predict road conditions. Zhu et al. [85] utilize a linear conditional Gaussian (LCG) Bayesian network model. Advantage of this model is its ability to work with continuous variables rather than discrete. However, learning of such model requires more and better data and therefore authors used data from macroscopic simulation. Paper written by Castillo et al. [7] describes the problem of predicting traffic flows and updating these predictions when information about origin-destination pairs and link flows becomes available. They propose to use Gaussian Bayesian network to deal with this problem. They, however, also work with synthetic data on a small problem. Shiliang et al. [65] are using combination of Pearson correlation coefficients, Gaussian Mixture Model (GMM) and the Competitive Expectation Maximization (CEM) algorithm to approximate their joint probability distribution of traffic flow in the traffic network. They are utilizing both the spatial and temporal information of the data from the small network in Beijing. Wang et al. [74] work with similarity of road segments. They use spectral clustering to find this similarity. Another work utilizing the BNs is the paper written by Herring et al. [27]. They use DBN for the prediction but instead of more usual traffic speed work with travel times. Their DBN has Gaussian observations and few hidden states. They also work with less complex data. Fei, Lu and Liu [79] proposed a method for forecasting travel time as the sum of the median of historical travel times, time varying random variations in travel time and a model evolution error. This is done by Bayesian inference-based dynamic linear model. Sun, Zhang and You [67] propose the use of Bayesian network to represent the joint probability distribution between the cause nodes (historical data) and the effect node (data to be forecasted). The resulting network is described as a Gaussian mixture model which parameters are estimated via the expectation maximization algorithm. The article [8] written by Castillo et. al. presents another BN approach to traffic modelling. They are using Beta-Gaussian BN to make a short-term traffic flow predictions. Hoflietner et. al. [28] try to model traffic in San Francisco by using Dynamic BN and fleet of floating cars. They also propose method to assess quality of the data from the floating cars. Perry and Hazelton [50] use Bayesian inference implemented by using Markov chain Monte Carlo methods to asses day-to-day traffic for various origin-destination (OD) pairs in the traffic network. The article written by Pascale and Nicoli is about another application of BN for the traffic modelling. They present their approach to automatic network adaptation for various types of the traffic. Wang, Deng and Guo [73] propose alternative approach to traffic modelling and prediction by combination of Bayesian statistics, Entropy-based grey relation analysis and Neural networks. Castillo, Menendez and Sanchez-Cambronero [69] use classical BN to predict OD pair flows. They are using OD pair matrix and not actual traffic network topology. In their other article Castillo et. al. solve similar problem by the application of Bayesian method to estimate OD matrices based on Gamma models. They decompose the algorithm into several consecutive optimization problems. Yet another approach is presented in paper written by Jiang and Fey [35]. They also use hidden Markov model (HMM) and use it in combination with neural network prediction. However, they use it on artificial data coming from the traffic simulation and are working with different kind of HMM.

I have chosen the following articles to represent current knowledge about traffic modelling with Kalman filters. Most of them use some form of LWR numerical model in their prediction phase. Wang and Papageorgiu [75] proposed the use of Extended KF to model and predict true state of the traffic. They argument, that standard KF approach does not represent well the nonlinearity of the traffic. The article [80] written by Xieng, Zhang and Ye proposes another improvement of KF. They are using basic KF but apply Discrete Wavelet Transform on the traffic data to remove corrupting noises. Yoon and Tchrakian [82] use slightly different Lagrangian LWR model. They also developed the special form of KF they named Regime based Kalman Filter. It is specifically made to handle results from the Lagrangian model. In the article [26] written by Guo, Huang and Williams it is proposed, that adaptive KF is very suitable for traffic modelling. Adaptive KF is form of KF with an adaptation mechanism for updating the process variances. The possibility of traffic flow modelling by the stochastic SARIMA + GARCH structure is also shown in the article. Work et. al. [76] suggest use of Ensemble KF for modelling the traffic. Ensemble KF is Monte-Carlo based KF and is explained in detail later. They also use discretized version of LWR, Cell transmission model. Coric, Djuric and Vucetic [71] have done similar work as Work et. al., but also propose that in case of true state reconstruction (not prediction) some smoothing technique can help to reconstruct true state of the traffic. They show several techniques ranging from piecewise linear interpolation to kernel regression.

4 Traffic data

Generally, data sources describing actual traffic situation on Czech motorways can be divided into two groups – stationary data sources and floating car data sources. Stationary data sources contain data provided by toll gates and the data provided by ASIM sensors. However, the value of the data from the toll gates is severely reduced by the fact, that they only contain information about large vehicles (trucks, buses, etc.). Data from toll gates thus only describes this specific part of the traffic, and is not usable for the description of general traffic situation. The following text briefly describes both ASIM sensor data and floating car data and summarizes their advantages and disadvantages.

4.1 ASIM Sensors

In the Czech Republic, the traffic situation is mostly monitored and evaluated using the stationary data. One of the most important source of stationary data is ASIM sensor network. ASIM sensors are placed on certain toll gates (all these toll gates are placed on the highways). They comprise of various sensors like passive infrared detectors and radars as can be seen in Figure 1. They are able to distinguish individual vehicle types, and measure their speed and intensity. Their measurements are aggregated every five minutes and mean speed and intensity are calculated.

They have number of advantages. One of the biggest advantages is the fact that there is no need for equipping vehicles with additional electronic devices. Consequently, speeds of all vehicles going through a sensor are measured and therefore they provide very accurate information. Another important advantage is detail of the data. ASIM sensors provide separate information about every lane of the monitored road. Moreover, since the ASIM sensor is able to distinguish type of the passing vehicle, it is only data source which is able to provide speeds and intensities for each type of the vehicle. Sample of the data can be seen in Figure 2.

There are, however some serious disadvantages of these sensors. In the Czech Republic, this network of measuring points is very sparse. There are only about 120 toll gates equipped with ASIM sensors and all of these are placed on the motorways. This low density is caused by related necessary expenses – installation of such measuring points is quite expensive. Also, they lack any flexibility. In case of some modification of the traffic network, they cannot be moved to more important or interesting place. There are also other limitations. Electronic toll gates divide roads into fragments of various length, some of them may extend to many kilometres. Thus, data obtained from ASIM sensors exactly describe only traffic situation around the tollgate.

4.2 Floating Car Data

The opposite to stationary data is Floating Car Data (FCD). The floating car data technique is based on the exchange of information between floating cars traveling on a road network and

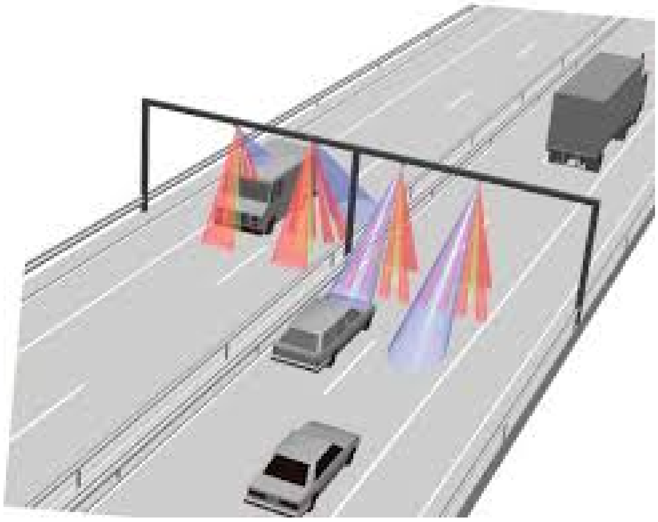

Figure 1: Structure of ASIM sensor [55]

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Begin	End	Count Total	Count Class 1	Count Class 2	Count Class 3	Count Class 4	Count Class 5	Count Class 6	Speed Total	Speed Class 1	Speed Class 2	Speed Class 3	Speed Class 4	Speed Class 5	Speed Class 6	
2	0:05	0:10:00	29	28	0	0	1	0	0	114	115	0	0	87	0	0	
3	0:10	0:15:00	24	21	1	0	2	0	0	106	108	100	0	91	0	0	
4	0:15	0:20:00	21	19	0	1	1	0	0	113	115	0	99	82	0	0	
5	0:20	0:25:00	26	24	2	0	0	0	0	116	118	97	0	0	0	0	
6	0:25	0:30:00	27	25	0	1	1	0	0	119	121	0	102	86	0	0	
7	0:30	0:35:00	37	34	1	1	1	0	0	113	115	90	92	88	0	0	
8	0:35	0:40:00	26	25	1	0	0	0	0	117	117	96	0	0	0	0	
9	0:40	0:45:00	24	24	0	0	0	0	0	126	126	0	0	0	0	0	
10	0:45	0:50:00	18	17	0	0	1	0	0	116	118	0	0	87	0	0	
11	0:50	0:55:00	17	15	2	0	0	0	0	119	123	95	0	0	0	0	
12	0:00	0:05:00	16	15	0	0	1	0	0	116	118	0	0	90	0	0	
13	0:55	1:00:00	27	20	3	3	1	0	0	114	121	104	89	92	0	0	
14	1:05	1:10:00	16	15	0	1	0	0	0	116	118	0	90	0	0	0	
15	1:10	1:15:00	21	20	0	0	0	1	0	113	114	0	0	0	98	0	
16	1:15	1:20:00	13	12	0	1	0	0	0	118	120	0	103	0	0	0	
17	1:20	1:25:00	20	19	0	0	1	0	0	111	112	0	0	81	0	0	
18	1:25	1:30:00	16	15	1	0	0	0	0	110	110	112	0	0	0	0	
19	1:30	1:35:00	14	14	0	0	0	0	0	120	120	0	0	0	0	0	
20	1:35	1:40:00	15	12	1	1	1	0	0	110	116	77	97	87	0	0	
21	1:40	1:45:00	15	15	0	0	0	0	0	118	118	0	0	0	0	0	
22	1:45	1:50:00	8	6	1	0	1	0	0	107	108	114	0	91	0	0	
23	1:50	1:55:00	21	18	1	0	1	1	0	111	113	95	0	92	104	0	
24	1:00	1:05:00	22	21	1	0	0	0	0	103	104	80	0	0	0	0	
25	9:55	10:00:00	43	39	0	2	2	0	0	118	121	0	98	82	0	0	

Figure 2: Example of ASIM data (data are from the RODOS system)

a central data system. The floating cars periodically send the recent accumulated data on their positions, whereas the central data system tracks the received data along the travelled routes. The frequency of sending/reporting is usually determined by the resolution of data required and the method of communication.

The most common and useful information that FCD techniques and ITS provide is average travel times and speeds along road links or paths [37, 84, 45, 83, 36]. They deploy FCD in order to predict short-term travel conditions, to automatically detect incident or critical situations [21, 72, 78], or determine Origin-Destination traffic flow patterns [64, 44].

Several FCD systems were presented, integrating short-term traffic forecasting based on current and historical FCD. However, these systems exploit data mostly from car flotillas to deliver real-time traffic speed information throughout large cities, signalized urban arterials or particular parts of traffic network, for example Italian motorway [13], Berlin [39], Beijing [41], Vienna [24], and many others.

The RODOS Transport Systems Development Centre[3] operates the system viaRODOS [4] which covers the whole traffic network of the Czech Republic [12]. Therefore, it is possible to monitor traffic situation from the global perspective. On the other hand, this globalization brings us several restrictions. To cover the whole traffic network, the system viaRODOS uses segmentation system which divides the highways and speed ways onto smaller parts - segments. The location code table from Traffic Message Channel (TMC) - a technology for delivering traffic and travel information to drivers - is used for identification of real world objects localization. Each row in location code table is strongly connected to a specific geographical entity (crossroads, roads, important objects, etc.). Locations used by TMC system are set by the rules for location identification defined by the international standard EN ISO 14819-3:1999. The location code tables for Czech Republic is created by Central European Data Agency (CEDA), which also addresses their certification on the international level.

As it was already mentioned, this approach is based on the measurements of location, speed, travel direction and time information from certain vehicles in the traffic. This information is obtained from the GPS receiver inside the car and broadcast by radio unit or cell phone. These broadcasts are collected and temporally and spatially aggregated by FCD engine, which then computes average speed for each part of the road in certain time interval (Figure 3). For example, D1 motorway is divided into the sections (TMC segments) with length from several hundred meters to few kilometres. The traffic speed is calculated each minute as a mean of speed of all floating cars that passed through the section in the last minute. Example of FCD data for one TMC segment can be seen in Figure 4.

This approach again has several advantages and drawbacks. Nowadays, an on-board unit including a GPS receiver becomes a standard equipment of corporate fleet cars. Moreover, increasing expansion of smart phones brought GNSS technology to my personal lives, where

[3]RODOS Transport Systems Development Center: `http://www.it4i-rodos.cz/defaultEN.aspx`
[4]viaRODOS is available on `http://www.viarodos.eu`

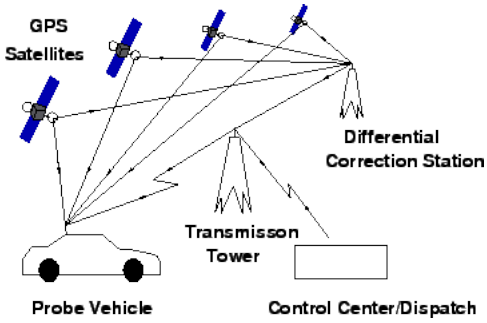

Figure 3: Scheme of FCD collection [31]

with combination of cheap connectivity, each vehicle can become a source of this type of data. The number of cars equipped with a GPS unit has doubled over the past five years. It can be expected that the trend will continue. It implies that the number of potential data sources will increase. Moreover, data from GPS receivers is not limited to the predefined places so the coverage is much larger than in the case of the stationary data.

	A	B	C	D	E	F	G
1	ID	Time Window	Free Flow Speed	Current Speed	Unique cars	Reliability	Date
2	0	0	103	83	20	80	3/24/2014 6:00:00 AM
3	0	1	103	85	18	82	3/24/2014 6:01:00 AM
4	0	2	103	92	10	89	3/24/2014 6:02:00 AM
5	0	3	103	85	17	82	3/24/2014 6:03:00 AM
6	0	4	103	89	13	86	3/24/2014 6:04:00 AM
7	0	5	103	99	4	96	3/24/2014 6:05:00 AM
8	0	6	103	103	0	100	3/24/2014 6:06:00 AM
9	0	7	103	103	0	100	3/24/2014 6:07:00 AM
10	0	8	103	103	0	100	3/24/2014 6:08:00 AM
11	0	9	103	102	1	99	3/24/2014 6:09:00 AM
12	0	10	103	93	9	90	3/24/2014 6:10:00 AM
13	0	11	103	91	11	88	3/24/2014 6:11:00 AM
14	0	12	103	95	7	92	3/24/2014 6:12:00 AM
15	0	13	103	88	14	85	3/24/2014 6:13:00 AM
16	0	14	103	91	11	88	3/24/2014 6:14:00 AM
17	0	15	103	97	5	94	3/24/2014 6:15:00 AM
18	0	16	103	102	1	99	3/24/2014 6:16:00 AM
19	0	17	103	103	0	100	3/24/2014 6:17:00 AM
20	0	18	103	93	9	90	3/24/2014 6:18:00 AM
21	0	19	103	102	1	99	3/24/2014 6:19:00 AM
22	0	20	103	102	1	99	3/24/2014 6:20:00 AM
23	0	21	103	100	3	97	3/24/2014 6:21:00 AM
24	0	22	103	93	9	90	3/24/2014 6:22:00 AM
25	0	23	103	96	6	93	3/24/2014 6:23:00 AM
26	0	24	103	95	7	92	3/24/2014 6:24:00 AM
27	0	25	103	94	8	91	3/24/2014 6:25:00 AM

Figure 4: Example of FCD data (data are from the RODOS system)

Disadvantages come mainly from the two sources: collecting of the data and the GNSS technology itself. The very same principle, that enables FCD to cover most of the traffic network can be its greatest fault. When some segment contains too few floating cars or none at all, quality of the output severely diminishes. This problem can easily happen even on the motorway during the off-peak hours and is quite usual on lesser roads. It drastically complicates usage of the FCD data from certain times and all the lesser roads. The other disadvantage comes from the GPS devices. As a part of GNSS technology, it can fail to provide precise outputs. The quality of outputs can be influenced by many factors such as the device quality, location, weather or other unpredictable and uncontrollable phenomena. All of this can have an impact on positioning, ranging from meters to tens of meters. Because GNSS is based on satellite technology, GPS receiver must be able to receive signals from several satellites. This can prove to be difficult in some cases. Typical example is an urban area with tall buildings which form obstacles between receiver and satellites. Then, GPS receiver is not able to report its position.

5 Traffic state prediction and proposed solutions

The objective of this study is to propose a macroscopic traffic state prediction model, which can be based on data sources mentioned in the previous section. This means that my prediction model can only utilize traffic speed. While using traffic intensity is technically possible, it would be very problematic as it is only available on ASIM sensors and their network is very sparse. Traffic density is missing altogether in my data. The utilization of both data sources also means that detailed study of the relationship between ASIM sensor data and FCD must be performed. This problem is dealt with by Granger causality [23] in Section 7.

Most straightforward solution to the traffic speed prediction problem is to solve it by application of some form of kinematic wave model or car following model. They accurately describe behaviour of traffic flow by application of physical models. Examples of this approach are articles written by Tang et. al. [70] [69] and Costesqua et. al. [11]. However, the absence of traffic intensity or density information in FCD and sparsity of other sensors rules out most of these traffic flow models, which are based these quantities. Even though there are some models which circumvent this problem (for example Cell transmission model - velocity (CTM-v), I still have insufficient data to exactly describe their initial and boundary conditions. While their results may seem realistic (as a dissolution of traffic jam from my CTM-v model shown in the Figure 5), their quality is quickly eroding with each iteration. Therefore, other prediction scheme for this problem must be proposed.

Other possible approach to traffic state prediction is to use historical data about traffic and relate them to the current state of the traffic by application of some statistical or machine learning approach. This can be described as a task of finding traffic state time series prediction

$$P_l = \underset{s \in S}{\operatorname{argmax}}\ p(s \mid x_1, \ldots, x_n),$$

where P_l is prediction of length l, S is set of possible predictions of certain length l, s is member of S and $x_1, \ldots, x_n$ are n last measurement of speed. Examples of this approach are described by Calvert et. al. [5] or Huang [30] and some other authors. This task is, however, also not a simple one. It is complicated by complex nature of the traffic data. They can, depending on type of their source, be inaccurate or not available in required quantity. Despite this fact, this approach is preferable in case of traffic state prediction on Czech Republic highways due to the unavailability of boundary condition data (i.e. there are not measurements on all ramps leading to and out of motorway). The solution was found in the form of graphical probability models. Based on the state of the art survey from the section 2, several approaches to this problem were chosen. However, they cannot be used without their modification and will require further research to be adapted to effectively utilize all of the available data sources.

These models include several probabilistic approaches which can all be interpreted as Bayesian networks(BN). BN are statistical models, which can represent a set of random variables and their

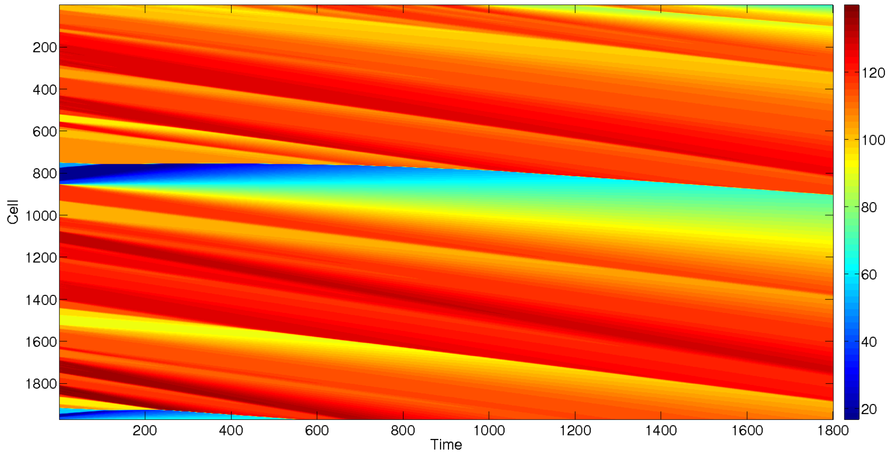

Figure 5: Result from simulation of D1 by CTM-v model (time is in seconds and color represents speed in km/h)

conditional dependencies via a directed acyclic graph. They can deal with the sparsity of the data via their ability to work with so called hidden variables. They also are capable of modelling uncertainty of input data. While general BN are not useful for time series prediction, they have many modifications like Dynamic Bayesian networks (DBN) and hidden Markov models (HMM) that can be readily adapted for TS predictions. The reasons for choosing the DBNs and HMMs as my probabilistic models is their handling of dynamic processes and ability to work with hidden variables. Despite having two data sources, it is impossible to observe every variable governing the traffic flow and this ability was the very reason for choosing these methods. The implementation of DBN would require to fulfil several tasks. Perhaps the most important one is to define structure of the graph representing network. Each vertex would represent the part of the road and its speed conditional probability distribution (CPD). The length of this stretch of the road will depend on the quality of the data. Some form of Expectation-Maximization (EM) algorithm will be implemented to handle learning procedure of the DBN to learn CPDs from the historical data. Algorithm for inference will be chosen and implemented depending on the final network structure.

Another possible approach is Ensemble Kalman filter (EnKF). EnKF model was, amongst other reasons, chosen because its ability to work with coarse physical model and to represent nonlinearity of the traffic. For the physical model, CTM-v was chosen because it is only physical model that requires only speed for the computation (and FCD contain only information about speed, not density or intensity). However, its application would require finding heuristics for boundary conditions (beginning and end of the modelled road and ramps). This heuristic will be based on analysis of historic data. Then this model will be implemented into Ensemble Kalman filter and resulting model will be tested and optimized on the historical data. EnKF are an algorithm that uses a series of measurements observed over time, containing noise and other uncertainties. It produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone.

Today's increase in volume of traffic is presenting yet another challenge to the traffic speed prediction. One of the products of this increase is growing number of traffic incidents. Predicting these incidents by standard methods presented above is very difficult as they are almost randomly distributed in the historical data. While it is possible to model free flow traffic (thanks to its periodicity and physical behaviour), it is vastly different problem to model traffic during the incident. There are many kinds of incidents from the traffic jam caused by road repairs to chain traffic accident which are also influenced by many factors which cannot be easily accounted for by use of historical data. However, these incidents also share many similarities in different times and places. Therefore, perhaps more useful information than the prediction of traffic speed is the prediction of duration of the incident. This can be formulated as

$$P_t = \underset{t \in T}{\operatorname{argmax}}\ p(t \mid x_1, \ldots, x_n),$$

where P_t is prediction of incident duration, T is set of possible predictions, t is member of T and $x_1, \ldots, x_n$ are n last measurement of speed. Therefore, predicting the state of the traffic during the incidents would require different approach than the one for the free flow prediction. There are some works, that have already been done in this field. Akira Kinoshita et. al. [1], for example, use probabilistic topic model. Ruimin Li et. al. [42] on the other hand use mixture models to compute traffic incident duration. Yangbeibei Ji et. al. [33] present slightly different approach to traffic incident length predictions in form of cell transmission model. I propose a different approach based on Markov chains, Dynamic Time Warping and Naive Bayes classification. This model would utilize incidents from the entire historical data set and relates the current incidents to them.

Theoretical foundations for all proposed methods can be found in Section 6 and their practical implementation and application within proposed algorithms in Section 8.

6 Methods

Theoretical foundations for the methods utilized in the developed algorithms are presented in this part. Note that, for the sake of space, only basic ideas and definitions are shown. More thorough descriptions and information about these methods can be found in the following literature:[47, 49, 20] for BNs, DBNs and HMMs, [17, 66, 16] for KFs, [10, 54, 53] for Evolutionary optimization and [77, 71, 43, 40, 25] for CTM-v. My algorithms also utilize other methods, but these are considered to be well known and their definitions are only mentioned through the citation of the relevant work.

6.1 Bayesian network

A Bayesian network is a graphical model that represents conditional independences between a set of random variables. Let us consider four random variables A, B, C , and D. The joint probability can be factored as a product of conditional probabilities:

$$P(A, B, C, D) = P(A)P(B|A)P(C|B, A)P(D|C, B, A).$$

This factorization, however, does not tell us anything useful about the joint probability distribution because each variable can be potentially dependent on every other variable. So let us consider the following factorization:

$$P(A, B, C, D) = P(A)P(B)P(C|A)P(D|C, B). \tag{1}$$

This factorization implies a set of conditional independence relations. A variable or set of variables X is conditionally independent from Y given Z if $P(X, Y|Z) = P(X|Z)P(Y|Z)$ for all X,Y and Z such that $P(Z) \neq 0$. With this factorization, it can be shown that given the values of B and C, A and D are independent:

$$\begin{aligned} P(A, D|B, C) &= \frac{P(A, B, C, D)}{P(B, C)} \\ &= \frac{P(A)P(B)P(C|A)P(D|C, B)}{\int P(A)P(B)P(C|A)P(D|C, B)dAdD} \\ &= \frac{P(A)P(C|A)P(D|C, B)}{P(C)} \\ &= P(A|C)P(D|B, C) \end{aligned}$$

A Bayesian network (BN) is a mean to represent a particular factorization of a joint distribution by a graph. Each variable is represented by a vertex in the graph. A directed edge is drawn from node X to node Y if Y is conditioned on X in the factorization of the joint distribution.

To represent the factorization (1) edge would be drawn from A to C but not from A to D. The Bayesian network representing this factorization is shown in Figure 6.

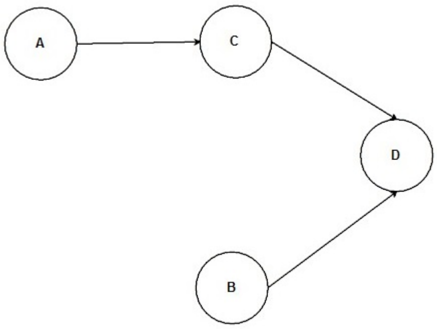

Figure 6: Example Bayesian network

It would be useful to state some definitions from the graph theory at this point. The vertex X is a parent of vertex Y if there is a directed edge from X to Y. Y is then called a child of X. The descendants of a vertex are therefore its children. A directed path from X to Y is a sequence of vertices starting from X and ending in Y such that each vertex in the sequence is a parent of the following vertex in the sequence. An undirected path from X to Y is a sequence of vertices starting from X and ending in Y such that each vertex in the sequence is a parent or child of the following vertex. This scheme defines basic semantics of a Bayesian network. Each vertex is conditionally independent from its non-descendants given its parents. Since each vertex represents a variable, I will often talk about conditional independence relations between vertices meaning conditional independence relations between the variables associated with these vertices.

If I generalize my previous statements, I can say that two disjoint sets of vertices X and Y are conditionally independent given Z, if Z d-separates X and Y. This means that along every undirected path between a vertex in X and a vertex in Y there is a vertex W with one of the following properties: either W has converging arrows and neither W nor its descendants are in Z, or W does not have converging arrow and W is in Z. It is therefore not difficult to infer many independence relations without explicitly using Bayes rule by visual inspection of the graphical model. From the BN in Figure 6 it is possible to see that A is conditionally independent from B given the set $X = C, D$, since $C \in X$ is along the only path between A and B, and C does not have converging arrows. It is, however, impossible to infer from the graph that A is conditionally independent from B given D.

It is worth to note that since each factorization contains a strict ordering of the variables, the connections obtained in this manner define a directed acyclic graph (i.e. graph with no cycles). Furthermore, there are many ways to factorize a joint distribution, and consequently there are many Bayesian networks consistent with a particular joint distribution. The absence of edges in a BN implies conditional independence relations. These relations can be exploited to obtain efficient algorithms for computing marginal and conditional probabilities. There is also one important rule associated with independence. Whenever a vertex is observed, then its parents become dependent, because they are both causes for explaining their child's value. This rule is known as explaining away. It is also important to mention, that not only network (or graph) topology is required. The prior probability of each state of a root vertex is also required for the purposes of preserving the computational ability.

One important property of the BN is that the graph can be considered as representation of the joint probability distribution for all variables. The tool to express this joint probability, as the product of the conditional probabilities that need to be specified for each variable, is called the chain rule:

$$P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|Par(X_i)),$$

where $Par(X_i)$ is the parent set of a node X_i . Then the conditional probabilities, the structure of the BN, and joint probability distribution can be used to determine the marginal probability or likelihood of each variable holding one of its states. This procedure is called marginalization.

The true power of BN is shown whenever one of these marginal probabilities is changed. The effects of the observation are propagated throughout the network. The probabilities of a various neighbouring node are updated in every propagation step. In case of simple BN the marginal probabilities or likelihood of each state can be calculated from the knowledge of the joint distribution, using the product rule, and the Bayes' theorem.

Therefore, to define a BN, I must specify the Conditional Probability Distribution (CPD) or pdfs of each node that has parents in addition to the prior probability of a root vertices. Since it is possible to assign evidence to every subset of nodes, I can define two types of reasoning that can be performed by BN:

- From known causes to unknown effects.
- From known effects to unknown causes.

There are two important actions that can be performed with BN. The first is called inference. One of great strength of BN is that I can infer conditional dependencies between variables by visually inspecting the graph. Therefore, set of BN vertices can be divided into non-overlapping subsets of conditionally independent nodes. This decomposition is a basic part of the probability inference. The inference is the task of computing the probability of each state of a vertex in a Bayesian network when the values of other vertices are known. The second step to perform inference is the belief propagation. Belief propagation is the action of updating the beliefs in each variable when observations of some other variables are given. Inference in BN can be done by 2 classes of methods. The first and the most usual one is the exact probability propagation in a singly connected network. To do this, I need to transform the network into singly connected structure. In the other class there are some approximate inference techniques (Monte Carlo inference techniques, inference by ancestral simulation, Gibbs sampling, Helmholtz machine inference,...).

The second one is called learning. It is usual that I do not know all of the conditional probabilities in the BN. Therefore, some learning techniques that enable us to complete the missing beliefs in the network must be employed. The role of learning is to find and adjust the parameters of the BN so that the pdfs defined by the network sufficiently describe statistical behaviour of the observed data. This procedure is usually realized by some form of EM algorithm.

6.1.1 Dynamic Bayesian network

Dynamic Bayesian Network (DBN) is a name of a model describing a system that is dynamically changing over time. DBN are usually defined as a special case of singly connected BN specifically designed for modeling the dynamic systems. All the vertices, edges and probabilities that make a static interpretation of a system are identical to a BN. Variables in DBN are called the state of a DBN, because they include a temporal dimension. These states satisfy the Markovian condition, which states that the state of a system at time t depends only on its previous state at time $t-1$. This property defines the First order Markov property: the future is independent of the past given the present.

The BN model is expanded by this fact. DBN allow not only connections within time slices, but also connections between time slices. These temporal connections introduce condition probabilities between variables from different time slices. These time dependencies are represented by the transition matrix which is called a Conditional Probability Table (CPT), since it represents the CPD in tabular form. Static CPDs can also be represented by CPTs.

As with static BN, the states of a dynamic one does not need to be directly observable. There might be influencing factors of other variables that can be directly measured or calculated. Moreover, the state of dynamic system does not need to be a unique and simple state. I may look at it as a complex structure of interacting states. Each state in a DBN at one-time slice may depend on one or more states at the previous time slice and on some states at the same time slice. So, generally speaking, in DBN the states of a system at time t may depend on system's states at time $t-1$ and possibly on current states of some other nodes in the fragment of DBN structure that represents variables at time t.

DBN can be described by probability distribution function on the sequence of T hidden-state variables $X = x_0, \ldots, x_{T-1}$ and the sequence of T observable variables $Y = y_0, \ldots, y_{T-1}$, where T is the time boundary for my model. This can be written as:

$$P(X,Y) = \prod_{t=1}^{T-1} P(x_t|x_{t-1}) \prod_{t=1}^{T-1} P(y_t|x_t) P(x_0),$$

where $P(x_t|x_{t-1})$ are state transition pdfs, $P(y_t|x_t)$ are observation pdfs that specify dependencies of observation nodes regarding to other nodes at time slice t; and $P(x_0)$ is an initial state distribution. The first two parameters had to be determined for all states in all time slices. Conditional pdfs can be time dependent or time independent. Time independent conditional pdfs can be parametric or nonparametric, when they are described using probability tables. A DBN can be discrete, continuous, or combination of these two depending on the type of the state space of hidden and observable variables. The similar tasks can be performed with DBN as with BN:

- Inference, estimation of the pdfs of unknown states given some known observations, and initial probability distribution.
- Learning, estimation of parameters of a DBN such that they best fit to the observed data, and make the best model for the system.
- Decoding, finding the best-fitting probability values for sequence of hidden states that have generated the known sequence of observations.
- Pruning, distinguishing which nodes are semantically important for inference in DBN structure, and which are not, and removing them from the network.

6.2 Hidden Markov models

A hidden Markov model (HMM) is a kind of a statistical Markov model in which the system being modelled is assumed to be a Markov process with unobserved (hidden) states (for more thorough description see [56]). A HMM can be presented as a form of dynamic Bayesian network. It can be modelled as a directed graph, where vertices represent random variables and edges represent dependencies. In opposite to simpler Markov models, the state is not directly visible to the observer. But the output, dependent on the state, is visible. Each state has a probability distribution over the possible outputs. Therefore, the sequence of outputs generated by HMM gives some information about the sequence of states. The name ’hidden’ refers to the state sequence through which the model passes, not to the parameters of the model. The model is still referred to as a ’hidden’ Markov model even in case if the parameters are known. General architecture of an HMM is shown in Figure 7.

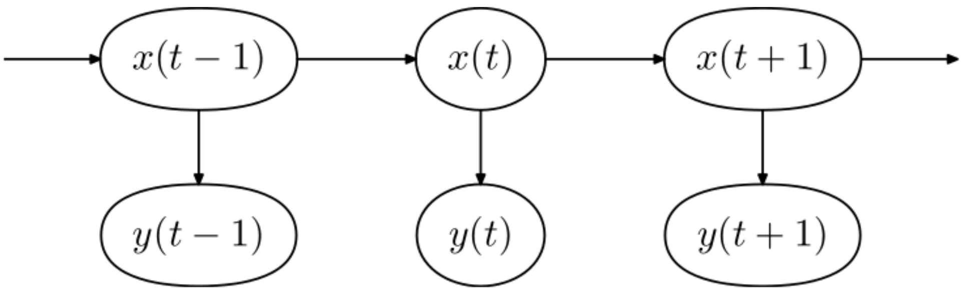

Figure 7: Snapshot of hidden Markov model

HMM structure comprises of two layers of vertices. Each vertex represents a random variable that can adopt any of a finite number of values. The random variables $x(t)$ represent the hidden states at time t. The random variables y(t) represent the emitted observation at time t. The oriented edges in the graph represent conditional dependencies. Figure 1 also shows, that the model has Markov property, i.e. the conditional probability distribution of the hidden variable $x(t)$ at time t, depends only on the value of the hidden variable $x(t-1)$, earlier states have no influence. In case of random variable $y(t)$, it again depends only on variable $x(t)$. In my case, I consider both values of x and y to be discrete, but emissions y can be modelled as continuous (using Gaussian distribution).

The HMM has three types of parameters: transition probabilities and emission probabilities and initial state probability. The transition probabilities determine the probability, that the hidden state $x(t)$ is chosen given the hidden state $x(t-1)$. The hidden state space is assumed to consist of N values which are modelled as a categorical distribution. Therefore, it is possible for each of the N possible states that are represented by a hidden variable at time t to determine a transition probability from this state to each of the N possible states at $t+1$. These transition probabilities are usually represented by a $N \times N$ matrix A called transition or Markov matrix. Rows of this matrix represent transition probabilities from the state denoted by row number to all other states (including itself) and naturally sum to 1. Emission probabilities are defined for each of the N possible states. They represent the distribution of the observed variable at a particular time given the state $x(t)$. The size and representation of these probabilities depends on the form of the observed variable. In my case the observed variable is discrete with M possible values, governed by a categorical distribution. Therefore, these probabilities can be represented by a $N \times M$ matrix B called emission matrix.

Initial state probability P is vector of length N and contains probabilities, that HMM begins in certain state.

HMM can perform many inference tasks. I am especially interested in two tasks: estimating the transition and emission probabilities for a hidden Markov model and finding probability of observing a certain emission sequence. The first task is to estimate matrices A and B given some emission sequence and their approximations A' and B'. This task can be performed by Baum-Welsh algorithm (description can be found in [6]). The second task is one of finding probability of observing a certain emission sequence $O = (o_1, o_2, \ldots, o_t)$ i.e. what is $P(O|\pi)$, where $\pi = (A, B, P)$ is the hidden Markov model. This task can be efficiently solved by the Forward-backward algorithm (description can be found in [6]).

6.3 Markov Chains

A Markov chain model can be described as a set of states denoted $S = s_1, s_2, ..., s_n$. The process starts in one of these states and at every time step it moves successively from one state to another. If the model is currently in state s_i, then it moves to another state s_j at the next time step with a probability of p_{ij}. This probability does not depend upon previous states of the model before the current state. This is called Markov property. The probabilities p_{ij} are called transition probabilities and are usually stored in $n \times n$ matrix called transition matrix. It is also possible for the process to remain in the state it is in with probability p_{ii}. Initial state is specified by an initial probability distribution. This can be done by specifying a particular state as the initial state.

Markov chains can be described by a sequence of directed graphs. The edges of n^{th} graph are labeled by the probabilities of moving from the state at time n to the other states at time $n+1$. This can be denoted as $\Pr(X_{n+1} = x \mid X_n = x_n)$. This information can also be represented by the transition matrix from time n to time $n+1$. However, Markov chains are usually assumed to be time-homogeneous.

The type of the Markov chain that I will use in this article is called an absorbing Markov chain. A state s_i of a Markov chain is called absorbing if it is impossible to leave it ($p_{ii} = 1$). A Markov chain is absorbing if it has at least one absorbing state, and if from every state it is possible to go to an absorbing state (not necessarily in one step). More information on Markov chains can be found here [9].

6.4 Naive Bayes Classifier

Naive Bayes classifiers are a simple probabilistic classifiers based on application of Bayes' theorem with independence assumptions between the features (more thorough description can be found in [48]). Naive Bayes is a simple concept: models are assign to problem instances, represented as vectors of feature values, where the model labels are drawn from some finite set of models. Basic principle connecting all naive Bayes classifiers states that the value of a particular feature is independent of the value of any other feature, given the model. Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers are proven to work quite well in many complex problems.

From strictly statistical perspective, naive Bayes is a conditional probability model. I am given a problem instance to be classified, represented by a vector of n features $\mathbf{x} = (x_1, \ldots, x_n)$. Classifier assigns to this instance probabilities

$$p(C_k | x_1, \ldots, x_n),$$

for each of k possible models.

However, this calculation may become unfeasible if the number of features n is large or if a feature can take on a large number of values. Therefore, some reformulation of the model is needed. The conditional probability can be decomposed by Bayes' rule as

$$p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

Because the denominator does not depend on C and the values of the x are given, the denominator is effectively scaling constant. By application of assumption of conditional independence between features I can rewrite equation for model probabilities as:

$$p(C_k|x_1,\dots,x_n) = \frac{1}{Q}p(C_k)\prod_{i=1}^{n} p(x_i|C_k),$$

where $Q = p(\mathbf{x})$ is a scaling factor. The naive Bayes classifier combines this model with some decision rule. The most common rule is to pick the hypothesis that is most probable. This rule is known as the maximum a posteriori decision rule (MAP). The corresponding Bayes classifier assigns a model label $z = C_k$for some k by following equation:

$$z = \underset{k\in\{1,\dots,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n} p(x_i|C_k). \tag{2}$$

6.5 Kalman filter

The Kalman filter is used to solve the general problem of estimating the state of a discrete-time controlled process that is defined by the linear stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}, \tag{3}$$

with a measurement $z \in \mathbb{R}^m$ that is

$$z_k = Hx_k + v_k. \tag{4}$$

The matrix A in the difference equation equation 3 relates the state at the previous time step to the state at the current step. The matrix B relates the optional control input to the state x. The matrix H in the equation 4 relates the state to the measurement z_k. The random variables w and v represent the process and measurement noise. They are assumed to be independent and with normal probability distributions.

$$P(w) \sim N(0, Q) \tag{5}$$

$$P(v) \sim N(0, R) \tag{6}$$

Q and R are the process noise covariance, respectively measurement noise covariance matrices.

The Kalman filter estimates a process by using a form of feedback control: the filter estimates the process state at some time and then obtains feedback in the form of (noisy) measurements. As such, the equations for the Kalman filter fall into two groups: time update equations and measurement update equations. The time update equations are responsible for advancing the current state and error covariance estimates in time to obtain the a priori estimates for the next time step. The measurement update equations are then responsible for the feedback. Their task is to incorporate a new measurement into the a priori estimate to obtain an improved a posteriori estimate. The time update equations can be considered as prediction equations, while the measurement update equations can be considered as correction equations. Scheme of the algorithm can be seen in Figure 8.

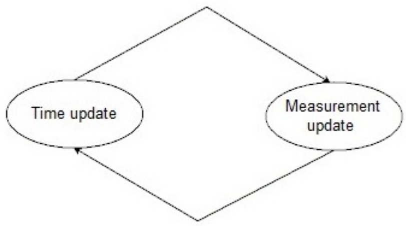

Figure 8: Kalman filter

The specific equations for the time updates are following:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

,

$$P_k^- = AP_{k-1}A^T + Q.$$

These time update equations project the state and covariance estimates forward from time step $(k-1)$ to step k. A is from equation 3 and Q is from equation 5. The specific equations for the measurement updates are:

$$K_k = P_k^- H^T \left(HP_k^- H^T + R \right)^{-1},$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-),$$

$$P_k = (I - K_k H)P_k^-.$$

H is from equation 4 and R is from equation 6. The first task during the measurement update is to compute the Kalman gain K_k. The measurement is incorporated after this step to obtain z_k, and then to generate an a posteriori state estimate. The final step is to obtain an a posteriori error covariance estimate.

After each time and measurement update cycle, the process is repeated with the previous a posteriori estimates used to predict the new a priori estimates. This recursive nature is one of the very appealing features of the Kalman filter because it makes practical implementations much more feasible.

6.5.1 Ensemble Kalman Filter

The ensemble KF (EnKF) was introduced as an alternative to the extended KF. Extended KF performance is poor when the state transition function is highly nonlinear. The EnKF belongs to a group of suboptimal estimators that use Monte Carlo or ensemble integration. The EnKF uses a collection of state vectors (called the ensemble members of system states) to propagate the state in time and to compute the mean and covariance needed for the correction step. The covariance estimated in this way is used to compute the Kalman gain. The correction step equation stays the same as in the traditional KF. The EnKF algorithm can be described by the following steps:

1. Generate N ensemble members of system states x_0^n by drawing N samples from a Gaussian distribution ($n = 1, 2, \ldots, N$).

2. Make a time update:
$$\hat{x}_t^n = A(\hat{x}_{t-1}^n$$

3. Compute the mean of the ensemble:
$$\mu_t = \frac{1}{N} \sum_{n=1}^{N} \hat{x}_t^n$$

4. Compute the covariance of the predicted state:
$$P_t = \frac{1}{N-1} E_t \left(E_t\right)^T \tag{7}$$
where $E_t = [\hat{x}_t^1 - \mu_t, \ldots, \hat{x}_t^N - \mu_t]$.

5. Calculate the Kalman gain:
$$K_k = P_t H^T \left(H P_t H^T + R\right)^{-1} \tag{8}$$

6. Obtain a new ensemble from the measurements:
$$x_t^n = \hat{x}_t^n + K_t[z_t - H\hat{x}_t^n] \tag{9}$$

7. Return to step 2.

6.6 Evolutionary algorithms

A typical feature of evolutionary algorithms is that they are based on working with populations of individuals. I can represent the population as a matrix $M \times N$ where columns represent the individuals. Each individual represents a solution to the current issue. In other words, it is set of cost function argument values whose optimal number combination I am looking for. One of early evolutionary algorithms can be found in [52]. The main activity of evolutionary algorithms is the cyclic creation of new populations which are better than the previous ones. Better population is a population whose individuals have better fitness. If I am looking for global minimum, I can schematically describe it with following formula:

$$\forall indA, indB \in \mathbb{R},\ f_{cost}(indA) < f_{cost}(indB) \Rightarrow F(indA) > F(indB)$$

where $indA, indB$ are population members, f_{cost} is cost function and F is function computing fitness (it is transformed cost function in my case). To create initial population, I need to define a sample individual (specimen). This specimen is also used to correct individual parameters which are out of valid range. I can define specimen as a vector of parameters. These parameters are described using three constants: type of variable (Real, Integer) and two boundaries that constrain value of parameter (Lo – Lower boundary, Hi – High boundary). Example given: Integer,Lo, Hi – Integer,1, 10.

$$Specimen = \{\{Real, \{Lo, Hi\}\}, \{Integer, \{Lo, Hi\}\}, \ldots, \{Decimal, \{Lo, Hi\}\}\} \tag{10}$$

Initial population P_0 is created using the specimen. Following formula guarantees that each individual of initial population will be randomly placed to space of possible solutions and its parameters will be inside of defined boundaries.

$$P_{i,j}^{(0)} = x_{i,j}^{(0)} = rand[0,1] \cdot (x_{i,j}^{(Hi)} - x_{i,j}^{(Lo)}) + x_{i,j}^{(Lo)},\ i = 1, \ldots, M,\ j = 1, \ldots, N, \tag{11}$$

where rand $[0, 1]$ represents function that generates random real number drawn from uniform distribution on the open interval $(0, 1)$ and $x_{i,j}$ is a j^{th} parameter of i^{th} individual. The goal is to find a vector $X = (x_1, \ldots, x_n)$ which minimizes my energetic function $f_{cost}(X)$ with regard to function and arguments restrictions. It is also necessary to ensure that all cost function argument values are from valid range while optimizing. I use very easy individual returning principle to manage that.

$$x_{i,j}^{\prime(EC+1)} = \begin{cases} rand[0,1] \cdot (x_{i,j}^{(Hi)} - x_{i,j}^{(Lo)}) + x_{i,j}^{(Lo)},\ if\ x_{i,j}^{\prime(EC+1)} > x_{i,j}^{(Hi)}) \vee (x_{i,j}^{\prime(EC+1)} < x_{i,j}^{(Lo)}) \\ x_{i,j}^{\prime(EC+1)},\ else \end{cases} \tag{12}$$

where $x_{i,j}$ is a j^{th} parameter of i^{th} individual and EC is a general expression that represents evolution cycle.

6.6.1 Differential Evolution (DE)

Differential evolution has proven to be an efficient method for optimizing real-valued functions. DE is trying to breed as good population of individuals as possible in loops called generations [53]. The very first step is defining constants affecting behaviour of evolution algorithm. Those constants are crossover probability (CR) which influences probability of choosing a parameter of the mutated individual instead of the original one, mutation constant (F) which determines volume of mutation of the mutated individual, population size (NP) and finally trial specimen. Now I have a specimen and I can create initial population vector X. When I have initial population prepared (called generation), loop begins. In each generation cycle, nested loop is executed. This loop is called evolution cycle. Each individual from current generation is bred to have better characteristic using mutations in evolution cycle. Algorithm finishes either if a solution is good enough (individual) or after defined number of generation loops had been executed. For my purposes, the DE/rand/1 algorithm mutation is the best choice. The notation DE/rand/1 specifies that the vector v to be perturbed is randomly chosen and the perturbation consists of one weighted vector.

$$v = x^G_{r1,j} + F \cdot (x^G_{r2,j} - x^G_{r3,j}), \tag{13}$$

where G is number of generation. Due to this mutation, new individuals are not affected by the temporary best individual from generation and space of possible solutions is searched through uniformly. More detailed description of Differential Evolution can be found in [54, 53].

Algorithm 1 Pseudo code (DE/rand/1)

procedure DE(X,f_{cost},NP,F,CR,$Specimen$)
    X: initial population (vector)
    f_{cost}: function returning fitness of current solution
    NP: population size
    F: mutational constant
    CR: Crossover threshold
    $Specimen$: trial specimen
    for do$I < Generation$ do begin
        for do$J < NP$ do begin Select J individual Select three random individuals from population Compute the new individual fitness Choose a better one from both individuals to new population
        end for
    end for
end procedure

6.7 Cell transmission model - velocity

CTM-v is based upon classical Lighthill-Whitham-Richards (LWR) partial differential equation. LWR PDE for modelling traffic on highways is:

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0 \tag{14}$$

where $q(x,t)$ is flow and $\rho(x,t)$ is density of the vehicles at location x and time t. This equation is derived from hydrodynamics theory and expresses the conservation of mass for a fluid of density ρ and of flux q. Empirical relation called fundamental diagram $q(x,t) = Q(\rho(x,t))$ is used for the expression of q as a function of ρ. Q is flux function and is considered to be independent of variables. To transform LWR PDE to velocity LWR (LWR-v) PDE, I will use the specific flux function called Greenshields:

$$v = v_{max}\left(1 - \frac{\rho}{\rho_{max}}\right)$$

where v_{max} and ρ_{max} denote the maximal velocity respectively the maximal density allowed by the model. This function expresses relation between density and velocity. Greenshields flux function can be easily inverted to express ρ as a function of v. Now I can substitute this expression into 14 to obtain LWR-v PDE:

$$\frac{\partial v}{\partial t} + \frac{\partial R(v)}{\partial x} = 0 \tag{15}$$

where $R(v) = v^2 - v_{max}v$. Variable change $v = v - \frac{v_{max}}{2}$ transforms 15 into its final form:

$$\frac{\partial v}{\partial t} + \frac{\partial v^2}{\partial x} = 0 \tag{16}$$

on the domain $(x,t) \in \langle a,b\rangle \times \langle 0,T\rangle$. The initial and boundary conditions are too long to be presented here and can can be found for example in [76]. For practical implementation, the LWR-v PDE is discretized using a Godunov numerical scheme[40]. By application of Godunov scheme, I obtain CTM-v. Let $N \in \mathbb{N}$ be number of time steps of length $\delta t = \frac{T}{N}$, $M \in \mathbb{N}$ number of space cells of length $\delta x = \frac{b-a}{M}$. Then v_i^n is called discrete value of v at time step n in cell i ($0 \leq n \leq N$ and $0 \leq i \leq M$). At each time step v_i^{n+1} is computed from the previous time step. This is done by formula:

$$v_i^{n+1} = v_i^n - \frac{\Delta t}{\Delta x}\left(g\left(v_i^n, v_{i+1}^n\right) - g\left(v_{i-1}^n, v_i^n\right)\right) \tag{17}$$

where the numerical flow function g is:

$$
g\left(v_1, v_2\right) = \begin{cases} R\left(v_2\right) & \textit{if } v_1 \leq v_2 \leq v_c \\ R\left(v_c\right) & \textit{if } v_1 \leq v_c \leq v_2 \\ R\left(v_1\right) & \textit{if } v_c \leq v_1 \leq v_2 \\ max\left(R\left(v_1\right), R\left(v_2\right)\right) & \textit{if } v_1 \geq v_2 \end{cases}
$$

where $v_c = \frac{v_{max}}{2}$, $R(v) = v^2 - v_{max}v$ and v_{max} is maximum speed allowed in the model. Note that Δt and Δx must satisfy Courant-Friedrichs-Lewy (CFL) condition:

$$
\left| \frac{\Delta t}{\Delta x} v_{max} \right| \leq 1
$$

For implementation of boundary conditions, I use ghost cells placed at each side of the domain defined by the boundary conditions to be satisfied.

7 Data Analysis

Intelligent Transportation Systems are highly dependent on the quality and quantity of road traffic data. The complexity of input data is often crucial for effectiveness and sufficient reliability of such systems. Generally, there are two types of data sources. The first data source is traffic information such as vehicle speed or traffic flow collected through fixed detectors placed along the road network at strategic points (in my case ASIM sensors). Other valuable data source is based on collecting traffic data through mobile devices and on-board units (FCD). Both of these data sources will be utilized in my algorithms and therefore several questions regarding their behaviour must be assessed. In computation of CTM-v model, I am using fundamental diagrams. Therefore, I must check whether my data sources follow these theoretical patterns. This task is, however, quite trivial due to the heavy restrictions imposed by information contained in my data sources. In case of ASIM sensors, I can only compare empiric traffic intensity to traffic speed. Due to the nature of the static sensors, there is no way how to measure traffic density. Visualization of the speed and traffic intensity relation from all ASIM measurements from March and April 2014 can be seen in Figure 9.

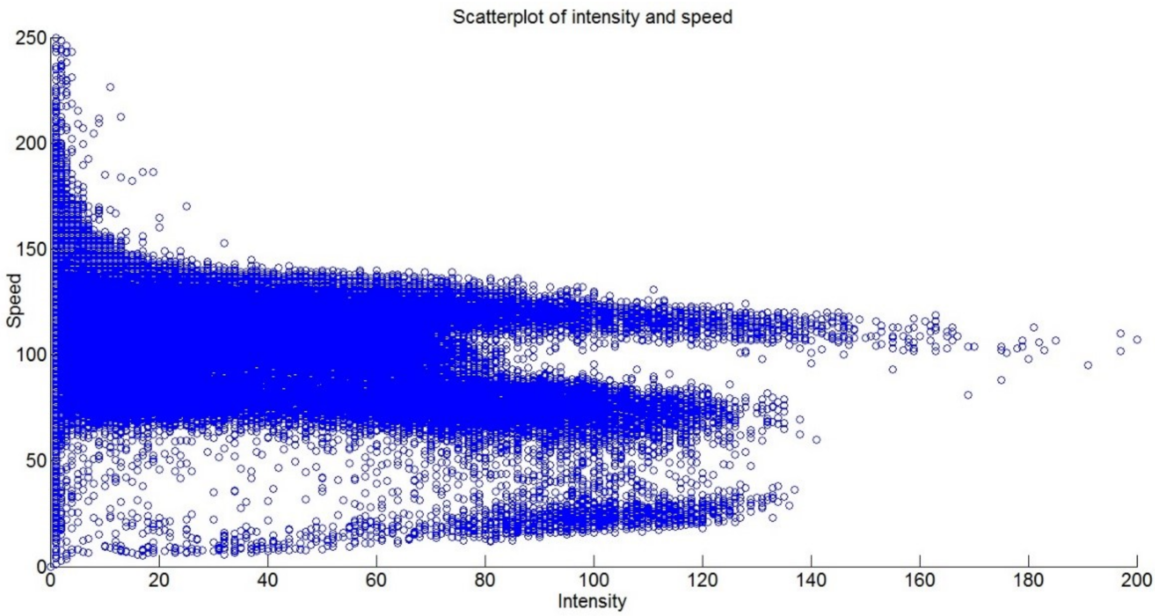

Figure 9: Scatter plot of intensity (veh/min) and speed (km/h) from ASIM sensors

When compared to the theoretical relation given by the fundamental diagram from Figure 10, ASIM sensor data roughly follow this theoretical pattern (although due to the aggregation, there are several characteristic shapes). Figure 9 also proves that larger part of the traffic follows uncongested flow pattern rather than congested one. It also misses its peak as traffic on D1 motorway is not dense enough.

In case of FCD, I only accurately know the speed of the traffic. While I know number of

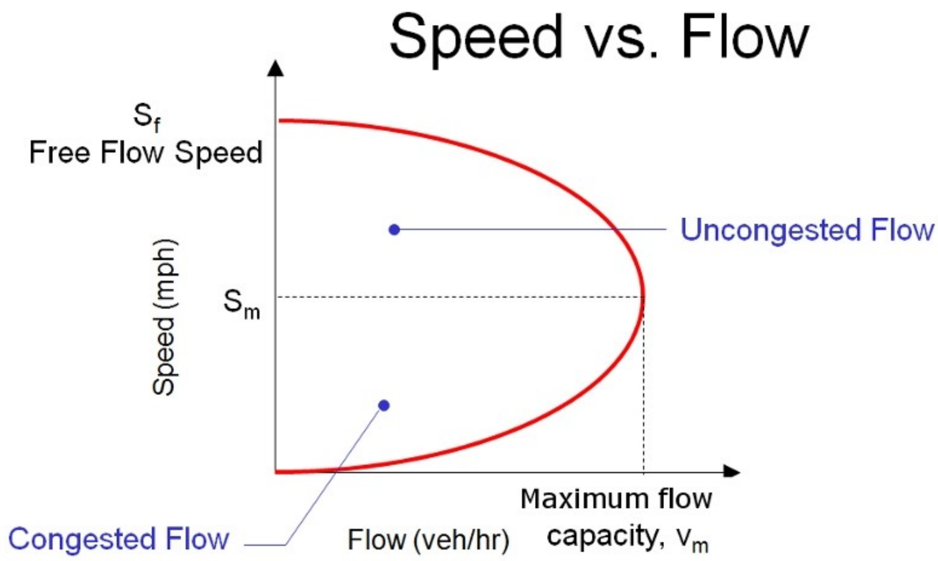

Figure 10: Theoretical plot of dependence of intensity and speed

unique FCD enabled cars on the segment (Figure 4) and have rough estimate of the penetration (usually around 10%), it is very difficult to determine their density due to the fact, that I have aggregated data. Therefore, density provided by the FCD is only very inaccurate estimate of the traffic density. Results of the comparison of this rough density estimate to the measured traffic speed from D1 motorway from the same time period as I did in case of ASIM sensors can be seen in the Figure 11.

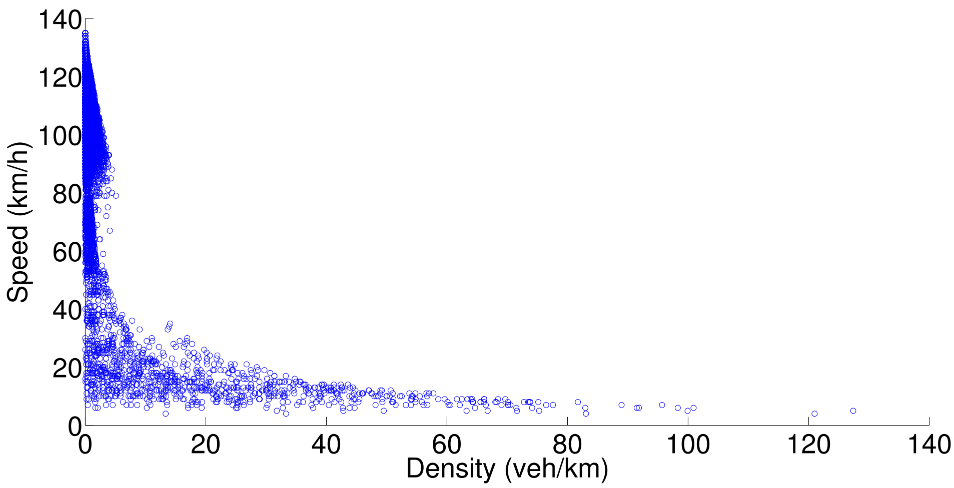

Figure 11: Scatter plot of density and speed from FCD

It is evident from the Figure 11 that some relation exists (i.e. scatter plot follows certain patterns and measurements are not randomly distributed). It roughly follows theoretical pattern that can be seen in the Figure 12 (not ideal linear model but one of the accepted nonlinear models depicted by green lines). This may be caused by the uncertain nature of the traffic density from the FCD.

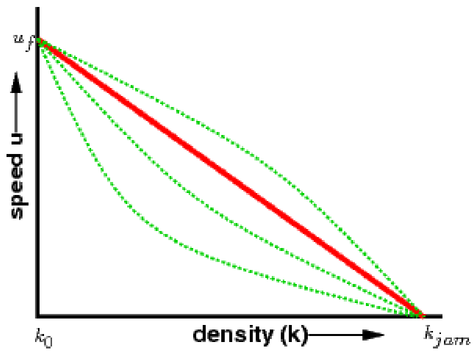

Figure 12: Theoretical plot of dependence of density and speed [32]

Taken together, data from both ASIM sensors and FCD roughly follows the theoretical patterns of the traffic quantities where the verification of these patterns was possible. In other cases, I only can suppose that this holds true. However, based on the previous two examples, this is in fact very likely so I can use these theoretical relations (like Greenshield's model) in my algorithms.

Another important analysis to be performed is figuring the relationship of FCD and ASIM sensor data. Because some of the sensors are working in a fixed place while the others are aggregating measurements over the specific part of the road. This configuration of data sources may or may not prove to be problematic and have to be thoroughly analysed.

Rest of this part is focused on finding relations between speed time series obtained from ASIM traffic profile detectors and floating car data with intention to use both data sources for traffic speed prediction. This research was published in[62]. The comparison was done by identifying Granger causality, which is presented in the next section. The experimental results are then described and analysed in Section 7.2.

7.1 Granger Causality

A variable x is said to Granger cause another variable y if past values of x help predict the current value of y. This definition is based on the concept of causal ordering [23]. Two variables can be correlated by chance but it is improbable that the past values of x will be useful in predicting y, given all the past values of y, unless x does in some way actually cause y. Granger causality is not identical to causation in the philosophical sense, but it does demonstrate the likelihood of such causation or the lack of such causation more forcefully than does simple correlation. For example, where the third variable drives both x and y, x might still appear to drive y though there is no actual philosophical causal mechanism directly linking the variables. However, there can be still Granger causality as one variable may be useful for predicting the other. The simplest test of Granger causality requires estimating the following two regression equations:

$$y_t = \gamma_{1,0} + \sum_{i=1}^{l} \gamma_{1,i} y_{t-i} + \sum_{j=1}^{l} \gamma_{1,l+j} x_{t-j} + \varepsilon_t \tag{18}$$

$$x_t = \gamma_{2,0} + \sum_{i=1}^{l} \gamma_{2,i} x_{t-i} + \sum_{j=1}^{l} \gamma_{2,l+j} y_{t-j} + \varepsilon_t \tag{19}$$

where l is the number of lags that adequately models the dynamic structure so that the coefficients of further lags of variables are not statistically significant and the error terms ϵ are white noise. Number of lags l is usually chosen using an information criterion, in my case Bayesian information criterion. If the l parameters $\gamma_{1,l+j}$ are jointly significant according to the F-test then the null hypothesis that x does not Granger cause y can be rejected. Similarly, if the l parameters $\gamma_{2,i}$ are jointly significant according to the F-test, then the null hypothesis that y does not Granger cause x can be rejected. This test is Granger causality test.

7.2 Experimental Results

Two toll gates with ASIM sensors and their appropriate FCD segments were chosen for the experiments. The first toll gate (labelled 34.1) is placed on 195.7th km of D1 motorway in direction to Ostrava and the other (labelled 15.2) is placed on 117.7th km of D1 in direction to Prague. Lengths of their corresponding FCD segments are roughly 2 km in both cases and the gates are placed in the middle of the segments. These gates and segments were chosen because of higher frequency of traffic incidents in these locations. These locations are also interesting because the first one is located near the city and the other one is not. Data comes from period of March to April 2014.

As it was mentioned in the introduction, my main interest is in determination of relationship between FCD and ASIM data also during the traffic incidents, respectively, whether this relation is causal or not. In case of longer time series without any incidents it can be quite unsurprisingly shown that there exists a causal relationship. This causality in case of gate 34.1 and its corresponding segment can be seen in Figure 13.

From the perspective of Granger causality test, both relations are causal (i.e. ASIM contains significant information about FCD and vice versa). In case of ASIM time series causing FCD time series, test statistic has value of 51.82 with critical value of 2.7 and in case of FCD causing ASIM, value test statistic was 12.36 with critical value 2.3. Therefore, in both cases I can reject null hypothesis that there is no causality between the time series. Comparable results were received in case of other sensors-segments and time periods.

Based on this I can declare that ASIM time series can be used for completion and prediction of FCD series and vice versa. It would not be difficult as, most of the time, traffic is periodic and easily predictable. This is, however, of little use as the most interesting part of the traffic are the

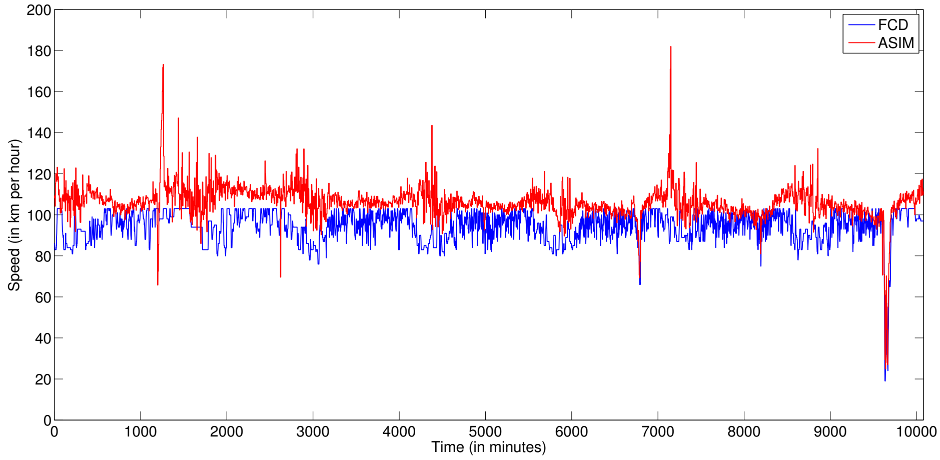

Figure 13: Time series of speed from ASIM sensor 34.1 and corresponding FCD segment form the third week of march 2014

Time of incident	ASIM causes FCD $[T/k]$	FCD causes ASIM $[T/k]$
8. 3. 2014 14:36-17:20	0.84/3.85	3.69/3.87
21. 3. 2014 16:19-17:55	6.49/3.89	9.7/3.04
16. 4. 2014 16:11-17:21	1.04/3.87	17.34/3.87
18. 4. 2014 16:49-18:48	0.51/3.87	19.07/3.1
23. 4. 2014 9:46-10:27	44.28/3.89	7.17/3.04
29. 4. 2014 17:00-18:33	18.51/3.87	5.92/3.88

Table 1: ASIM sensor 34.1 causality

traffic incidents. It is quite possible that margin of error allowed by the Granger causality can come from these rare events. Therefore, it is important to know whether these causal relations work even in case of traffic incidents on shorter time series. There were 16 incidents detected by my method during observed period (six in case of sensor 34.1 and ten in case of sensor 15.2). Results from Granger causality tests performed on these incidents can be seen in Tables 1 and 2 (T is test statistic and k is critical value; tests which have not confirmed causality are marked by red colour).

Three different kinds of outcome can be seen in Table 1 and Table 2. The first one is that both causality tests were passed. This is the most usual outcome and it implies that both ASIM data and FCD data react on the incident at roughly the same time. It probably means that incident happened somewhere near the toll gate with ASIM sensor. Such incident can be seen in Figure 14. The fact that this outcome represents more than a half of the detected incidents means, that most of the time, both of the sensors react well and at the similar time on the development of the traffic.

The second possible outcome is that there is only one-way causality between the time series.

Time of incident	ASIM causes FCD $[T/k]$	FCD causes ASIM $[T/k]$
24. 3. 2014 9:17-21:05	14.02/3.85	21.75/3.85
25. 3. 2014 7:13-12:15	5.14/3.02	15.89/3.86
25. 3. 2014 17:05-21:07	16.73/3.01	0.63/3.86
26. 3. 2014 9:01-10:16	1.55/3.87	0.7/3.87
26. 3. 2014 14:13-15:05	0.16/3.92	3.42/3.93
26. 3. 2014 15:36-20:17	7.35/3.88	3.03/3.02
28. 3. 2014 15:55-19:16	0.88/3.86	5.24/3.86
1. 4. 2014 16:05-16:46	8.41/3.89	9.71/2.05
10. 4. 2014 16:40-17:33	14.07/3.87	16.20/3.02
22. 4. 2014 7:30-9:56	12.89/2.40	9.52/3.88

Table 2: ASIM sensor 15.2 causality

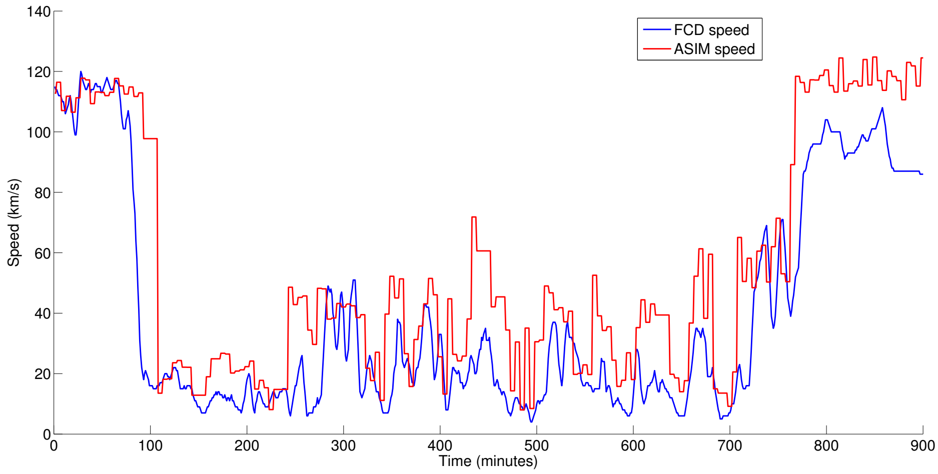

Figure 14: Time series of speed from ASIM sensor 15.2 and corresponding FCD segment form 24. 3. 2014

Such example is in Figure 15. It happens either when the incident is further from the toll gate with ASIM and is sooner registered by FCD or is ahead of the ASIM sensor. Due to the sparsity of the ASIM sensor network (sensors on D1 are placed about 12 km apart), it is much more usual that FCD data detects the incident sooner than ASIM sensors and are therefore usable for the predictions of speeds in ASIM sensor time series. Opposite scenario is much rarer. If I consider the fact that most of the time both series are roughly equivalent, I can assume that it is quite safe to predict ASIM speed time series from FCD time series. If I try it other way, there is greater possibility of failure due to the sensor sparsity.

The third possible outcome is that there is no causality either way. These outcomes prove to be problematic because they cannot be easily explained and can imply some problems regarding predictability. Because of this fact, they will be thoroughly analysed.

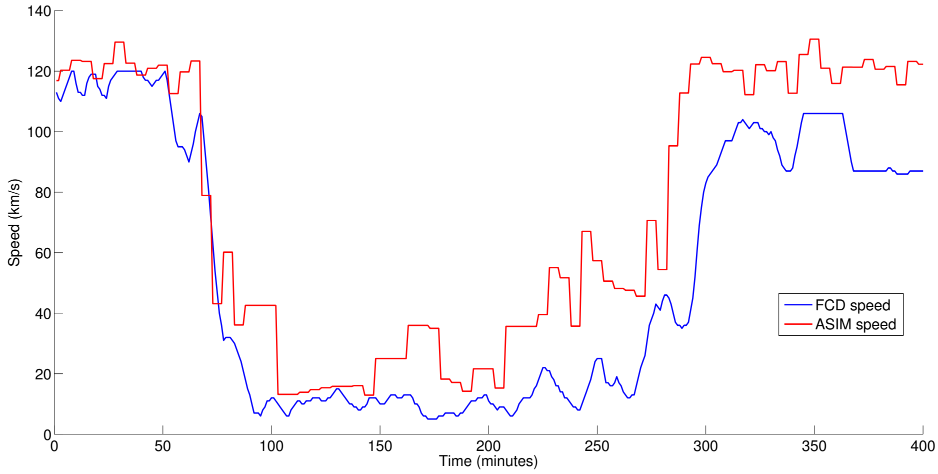

Figure 15: Time series of speed from ASIM sensor 15.2 and corresponding FCD segment form 25. 3. 2014

The first of the three analysed incidents is the incident from 8.3.2014 from 195.7th km of D1 in the direction to Ostrava. Time series from FCD and ASIM are shown in Figure 15. The figure shows an apparent disparity in the data between FCD and ASIM data. This is a very busy part of the D1 motorway. This FCD segment is placed between major motorway ramps. One ramp leads to the motorway to Vienna and the other one to motorway to Bratislava. The segment is approximately 2300 m long and ASIM sensor is installed 1400 m from the beginning of the segment.

There was short-term closure of the left lane realized on this segment on 8.3.2014. This meant a reduction in the number of lanes. Beginning of this short-term closure was on 195.2 km and end of this closure was on 195.6 km, approximately 700 meters before the ASIM sensor. The short-term closure was implemented between 3:00 p.m. and 6:00 p.m. A traffic column was created in the front of the closure, which is visible in the FCD data in Figure 16. However, data from ASIM sensor show no traffic problem. This is because the column was created in front of traffic restrictions, but ASIM sensor is placed behind the restriction where the traffic flow is not restricted. Despite relative closeness of the incident to the ASIM sensor, it has failed to detect it. This situation confirms fact that to predict FCD time series from ASIM can be inaccurate. On the other hand, the FCD data source can be used to monitor the situation.

On 24. 3. 2014 at 9:00 at 112th km started the modernization of the D1 motorway. The repaired section started at 112th km and ended at 104th km in the direction to Prague. In the period between the 24.3. and 30.3. 2014, the traffic was restricted to one lane. Such traffic engineering measures at places with so intense traffic always cause problems and so it was in this case. Both traffic incidents with no causality happened during this period. Bottleneck, the place where two lanes were merging into one was placed at 112.5th km. Beginning (or more exactly the end) of TMC segment was distanced approximately 3500 m against the road direction (116th

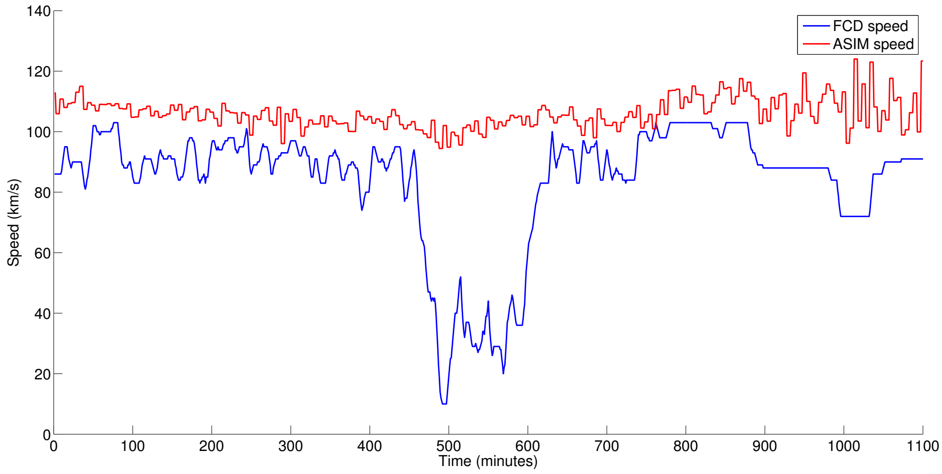

Figure 16: Time series of speed from ASIM sensor 34.1 and corresponding FCD segment form 8. 3. 2014

km) from the bottleneck and ASIM sensor was placed additional approximately 1500 meters from the beginning of the segment. This TMC segment was about 3500 meters long.

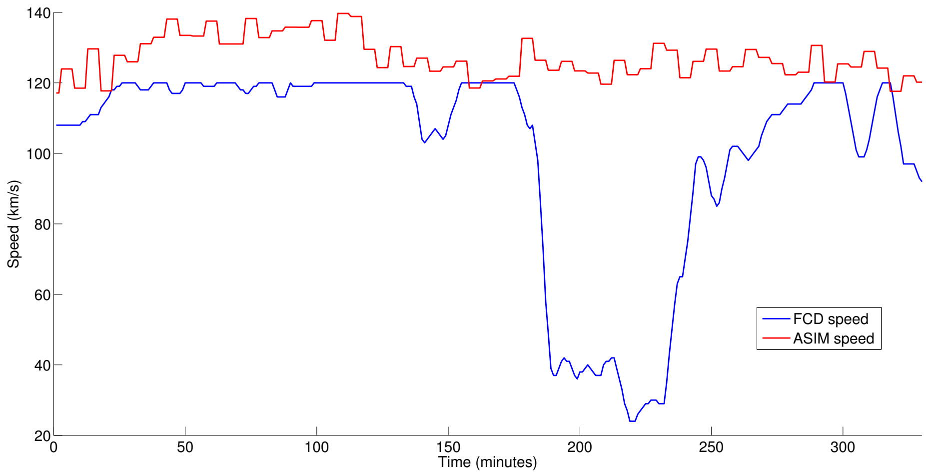

Figure 17: Time series of speed from ASIM sensor 15.2 and corresponding FCD segment form 9:00 26. 3. 2014

The exported data from FCD and ASIM sensor show that in the first and the second day of reconstruction (24. 3. and 25. 3.) there were long traffic columns with length of more than 5 km almost all day in the front of the merge. In those days, the speeds from FCD data and ASIM sensor were almost identical. This phenomenon was caused by the very rapid formation of traffic columns, which reached the ASIM sensor within minutes. The next day (26. 3.), it is already evident that after a series of media reports a certain percentage of drivers chose to postpone their trip, or chose another type of transport. This fact caused formation of shorter columns and their slower formation, which is evident in the difference between the speed of FCD and ASIM sensor (see Figure 17). Because the column has started reaching FCD segment, but has not reached the ASIM sensor, the speed of the FCD reduced, but the monitored detection sensor was still recording the speed of freely moving traffic flow. This is evident from the incident that happened on 26. 3. 2014 between 9:00 and 10:00 when the column intervened into the FCD segment, but has not reached the measured profile. In case of the next incident (14:00 on the same day), the column reached ASIM sensor yet.

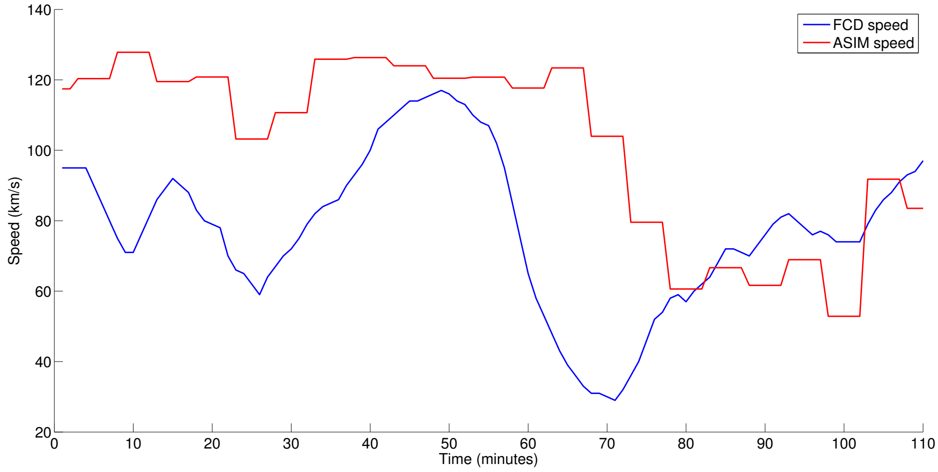

Figure 18: Time series of speed from ASIM sensor 15.2 and corresponding FCD segment form 14:00 26. 3. 2014

From Figure 18 there is apparent delay in detection on the column by ASIM sensor. This is caused by slow formation of the column. The same is true in case of the dissolution of the column. The speed at first rises in ASIM sensor measurements and then gradually increases in FCD data. Only at the moment when the column does not interfere with the measured segment, traffic speeds again start to match. This again proves that prediction of FCD time series from ASIM is problematic due to the many specific configurations where profile sensor fails to detect traffic incident.

From the obtained results, two facts are evident. The first is, that in case of unconstrained traffic flow, both data sources can be used safely together for prediction. In case of constrained traffic flow, things get more complicated. It is evident that in more than a half of incidents, there is Granger causality between FCD and ASIM data in both ways. In the other cases, there was only one-way causality. It was mostly due to the fact, that when the incident happens some distance from the ASIM sensor, FCD is usually much swifter to record this incident than ASIM. Sometimes there is no causality at all between these time series but by thorough analysis of these incidents, I have proven that again this is caused by static nature of ASIM detectors which fails to detect some phenomena that can happen even in its close vicinity. Therefore, I can summarize that it is quite reliable to predict values of ASIM time series from FCD during the traffic incidents. Predicting FCD from ASIM can work most of the time too but is nowhere as reliable due to the nature of the sensor. Therefore, every time I am using these two data sources together, physical configuration of both sensors must be checked, or I may receive wrong results, especially in case of traffic incident or traffic jam. This fact was taken in mind in designing the experiments performed in this research.

Results of this analysis underline importance of FCD data in Czech motorway traffic. ASIM sensors provide valuable information, but due to their sparsity are not able to describe accurately traffic in its entirety. They are failing especially when the traffic column is not long enough to reach the ASIM sensor or in the areas with many ramps. Due to this fact, it is very important to incorporate FCD data to any traffic model for Czech motorways, especially one that want to describe traffic during traffic jams or incidents. The intelligent combination of FCD with on-road sensors represents the perfect inputs to dynamic traffic models.

8 Algorithms

Developed algorithms and their experimental results are presented in this section. Three of these algorithms are for the general traffic speed time series prediction. The last one is designed specifically for the classification and prediction of traffic incident length. All algorithms have in common that they were implemented in MATLAB programming language and utilized rudimentary parallelization (mostly parallel for cycles). While MATLAB programming language is known for its slowness, it contains many tools that ease development like numerous visualization methods and easy debugging. For the real deployment, these algorithms will be implemented in some more performance orientated language like C++ or R.

8.1 Dynamic Bayesian networks approach

8.1.1 Algorithm

Dynamic Bayesian networks based algorithm will be presented in this part. This algorithm was published in[59]. Proposed algorithm is based on use of both historical and current traffic data and predicts traffic speed for several consecutive segments of the road (FCD segments are used in this work). These data are represented by speed time series coming from the two available sources: ASIM sensor network and FCD. Both of these sources have important roles in my model. ASIM sensors provide reliable information but, as a standalone information source, suffer from the two disadvantages. The first is spatial one, i.e. their network is sparse and is not covering equidistantly the motorway. The second is temporal one as the ASIM sensors have longer aggregation period and therefore are not optimal for on-line prediction. This problem can be fixed with FCD, which are available for all sections of the Czech Republic motorways and have much shorter aggregation period. However, these data can be quite inaccurate due to the lower penetration of the traffic by floating cars or other technical issues. Therefore, ideal prediction algorithm can utilize both the accurate but localized ASIM sensor data and widely available FCD, which suffer from the fact that they only measure speed of part of the traffic flow.

One of the possible solutions for this problem was found in the form of dynamic Bayesian network also called coupled hidden Markov model. The example structure of the network can be seen in Figure 19. Each vertex represents segment of the road. Edges are drawn in such way that future speed value v^s_{t+1} of the segment depends on its last value v^s_t and last values of its neighbouring segments v^{s-1}_{t-1} and v^{s+1}_{t-1}. Hidden states represent true state of the traffic described by ASIM sensor data and measurements are observed noisy data represented by FCD. Both of these data sources are used during the learning of the network. One of the advantages of this model is that parameters of the model can be learned even when there are no ASIM sensors on some of the segments i.e. some of the hidden states are not observable at all. When properly trained, this network is able to predict the true state of the traffic speed (i.e. hidden state) from the n last values of FCD measurements.

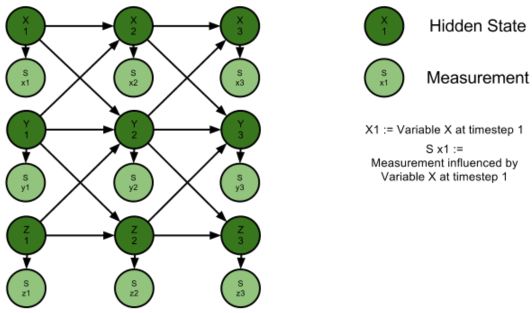

Figure 19: Snapshot of DBN [46]

General progress of my algorithm can be summarized into following steps:

1. Learning data preprocessing

2. Learning of the networks

3. On-line prediction

Let us go through individual steps of the algorithm more thoroughly. In the first step, ASIM data and FCD are pre-processed for use in my network. This means that both of these continuous variables have to be transformed to discrete ones with m and n values for hidden and measurement nodes respectively. These numbers must not be too high, because the number of parameters in such model rises steeply (model requires $N \cdot m + N \cdot n + N \cdot n \cdot m + (n-2)^4 + 2n^3$ parameters) and it may not be possible to learn all these parameters in reasonable time. Some other works also use continuous observations (measurements) but I have decided to utilize simpler method to ease the learning of the network. The discretization is usually done by rounding the speeds to a certain number (i.e. 5 km/h). However, as it is evident from the Figure 20, most of the measurements of ASIM sensors are distributed between 80 km/h and 130 km/h. Therefore, such rounding would produce few states with reasonable probability and greater number of improbable states. To address this issue, discretization based on probability density is proposed. State discretization will be smoother in areas with higher probability density and coarser otherwise. For example, if the speed is less than 60 km/h, it would be rounded to nearest multiple of 20 km/h and if the speed is higher than 60 km/h, it would be rounded to nearest multiple of 5 km/h. This procedure should result in states with reasonable probability. The data are also divided into 4 portions (data from 6:00 to 10:00, data from 10:00 to 14:00 and data from 14:00 to 18:00 and the rest, which is not used for prediction). This is done because the traffic speed time series are highly non-stationary and therefore creating only single model for all day would produce remarkably worse results. These periods are also naturally identifiable as morning peak, noon off-peak and afternoon peak.

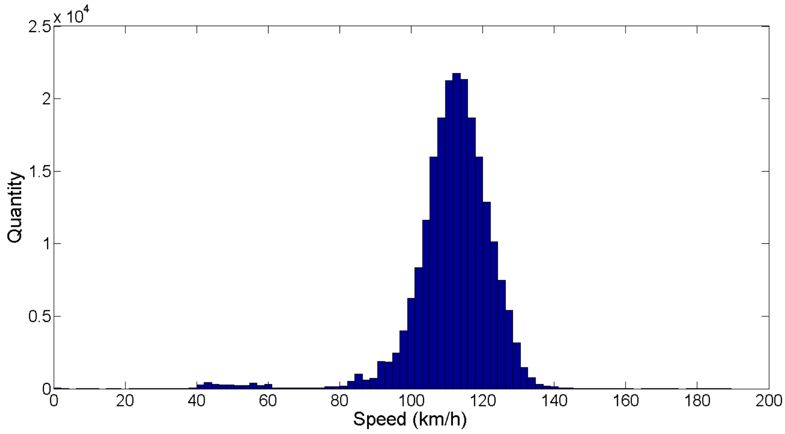

Figure 20: Histogram of traffic speed from D1 motorway

The next step is the learning procedure of my model. This can be described as

$$\theta^* = \arg\max_{\theta} \log P(Y|\theta, C), \tag{20}$$

where θ^* are learned CPDs, θ are randomly initialized CPDs, Y are historical data for both ASIM and FCD and C is network structure of my DBN. In case of complete data (i.e. modelled part of the road has both ASIM sensors an FCD available for all sections), this equation can be computed by classical maximum likelihood estimation. However, more common scenario is, that some sections does not have ASIM sensors and therefore MLE does not decompose into a sum of terms a transfers Equation 20 into

$$\theta^* = \arg\max_{\theta} \sum_{Z} logP(Y, Z|\theta, C),$$

where Z are unobserved variables (i.e. sections without ASIM sensor). Therefore, EM algorithm [49] must be utilized to perform this learning task. EM algorithm is ideal for this problem because it can handle incomplete or unobserved data.

In the last step of my algorithm, special inference task called prediction is performed. In terms of my network, it means finding:

$$P(x(i, t + T)|y(:, 1 : t), C, \theta^*)$$

where $x(i, t+T)$ is hidden state of node i at time $t+T$ and $y(:, 1 : t)$ represents measurements from the time period foregoing the prediction. Predicted value for each hidden variable is then calculated by maximum a posteriori decision rule (MAP) as:

$$z = \arg\max_{k \in \{1,\ldots,m\}} P_k(x(i, t + T)|y(:, 1 : t)) \tag{21}$$

where $P_k(x(i, t+T)|y(:, 1:t), C, \theta^*)$ is the probability of hidden variable i at time $t+T$ be at state k. This approximate inference is performed by The Boyen-Koller algorithm [4]. Pseudo code for the algorithm is written in Listing 2.

Algorithm 2 DBN prediction algorithm

procedure DBN PREDICT(X, Y, Yc, θ, T)
  X: historical ASIM values
  Y: historical FCD values
  Yc: current FCD time series
  θ: initial guess of CPDs
  T: length of the prediction
  $[Y_p, X_p, Yc_p]$=Preprocess(X,Y,Yc); ▷ Preprocess learning and prediction data
  θ^*=EMlearning(θ,X_p,Y_p); ▷ EM algorithm learning
  pd=BoyenKoller(Yc,θ^*,T); ▷ Approximate inference
  p=MAP(pd); ▷ Prediction based on MAP
end procedure

8.1.2 Experimental results

My algorithm was implemented in Matlab environment and utilized basic parallelization to accelerate the computation. From the performance perspective, the only problematic part is the learning of parameters which can easily take tens of minutes. However, given enough historical data, it is not necessary to frequently update the model. The computation of prediction with trained model takes only tens of seconds.

The algorithm was tested on data coming from Czech D1 motorway from the period of 1.9.2014 to 30.11.2014. I have chosen 3 consecutive sections from 180th to 195th km in the direction of Brno (i.e. graph structure was the same as in Figure 19). Traffic speeds from both ASIM data and FCD were discretized according to the scheme presented in the previous part and had 18 and 26 states respectively. Each of these sections contained ASIM sensor. Two of these were used for learning and the values from the last one was not included in training set (i.e. remained unobserved). It was, however, used for result verification. I have used data from September to October for training and data from November for evaluation. Root Mean Square Error (RMSE) was chosen as a measure for quality of the prediction. All three learned models were tested and gave comparable results. For the sake of space, I only present results from the first one. I have run online prediction predicting the state of the traffic speed in these three segments 5 minutes ahead for the duration of one hour. Results from one of these predictions can be seen in Figures 21, 22 and 23.

The predictions for the first segment (Figure 21) have RMSE of 7.83, for the second segment (Figure 22) RMSE of 7.90 and for the third which was the one without the ASIM in learning set (Figure 23) RMSE of 10.66. In terms of RMSE mean value from all predictions, the confidence

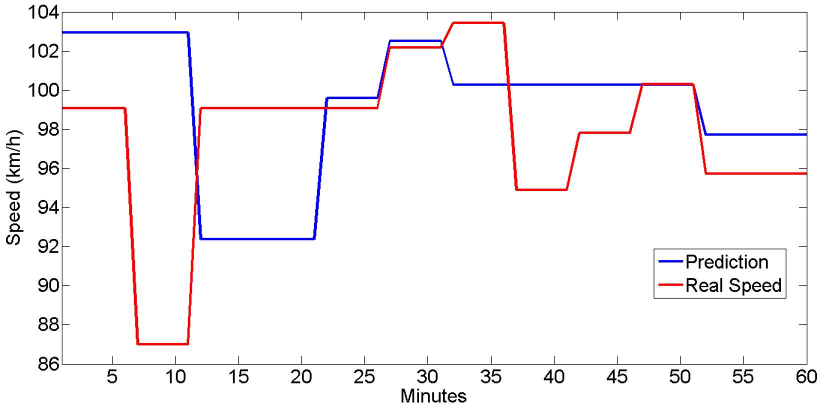

Figure 21: Traffic speed prediction for the first segment with RMSE of 7.83

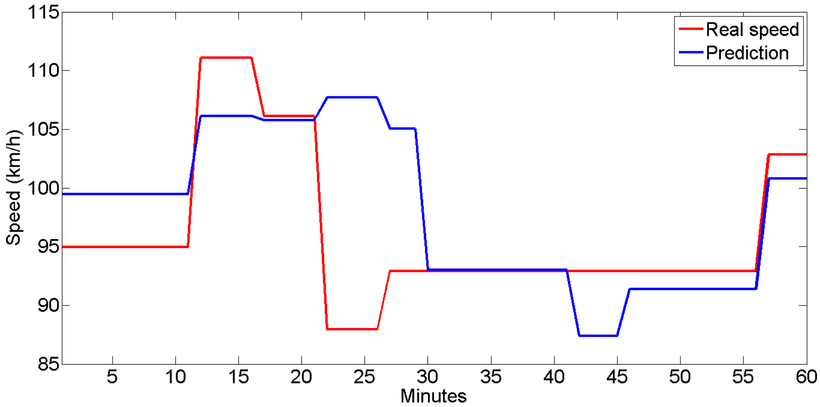

Figure 22: Traffic speed prediction for the second segment with RMSE of 7.90

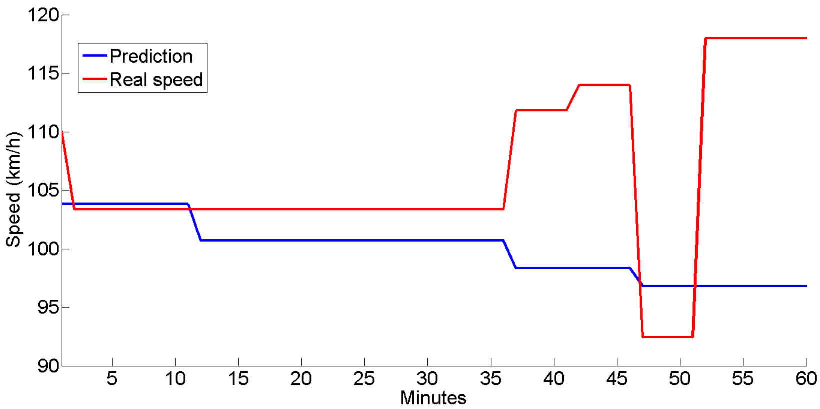

Figure 23: Traffic speed prediction for the third segment with RMSE of 10.66

interval for the first segment is $< 8.04, 8.39 >$, for the second is $< 8.72, 9.68 >$ and for the third is $< 12.41, 12.56 >$. Results from the third part are clearly influenced by the presence of the hidden variable. Solution to this problem is discussed in the Conclusion.

8.2 Hidden Markov model approach

8.2.1 Algorithm

Now let me present my Hidden Markov model based algorithm. This algorithm was published in[61]. As it has already been mentioned, my algorithm is based on use of both historical and current traffic data. These data are represented by speed time series coming from the both available sources: ASIM sensor network and FCD. The algorithm predicts speed for one segment of the road. Due to the scarcity of ASIM sensors and occasional problems with their location, it is perhaps better to predict the traffic speed based on the FCD, which are available for all sections of the Czech Republic highways. However, these data can be quite inaccurate due to the lower penetration of the traffic by floating cars or other technical issues. Therefore, this prediction algorithm can utilize both the accurate but strongly localized ASIM sensor data and widely available FCD, which suffer from the fact that they only measure speed of part of the traffic flow. As it is true for the most prediction algorithms, this algorithm is capable of predicting the traffic speed by up to the 30 minutes. Quality of longer predictions seriously deteriorates.

One of the possible solutions for this problem was found in the form of hidden Markov model. Both data sources can be utilized in the framework of HMM. ASIM sensor data can be utilized for training of the model and roughly represent the hidden state sequence. The prediction is then done on the FCD, which are regarded as emission. General progress of my algorithm can be summarized into following steps:

1. Create approximation of transition and emission matrices.
2. Train HMM on historical data from the FCD and approximation matrices.
3. Generate n predictions (emissions) and compute their likelihood.
4. Choose m most probable sequences and compute their weighted mean.

Let us go through individual steps of the algorithm more thoroughly. In the first step, ASIM data are utilized to create approximated transition matrix $A^{'}$ from the nearest ASIM sensor to the selected FCD section. It is defined as:

$$A^{'} = \{a^{'}_{ij}\} = P(X_t = j|X_{t-1} = i),$$

where X_t is hidden random variable and $P(X_t|X_{t-1})$ is independent of time t (Markov property). This matrix is calculated like transition matrix for the standard Markov chain from the historical data (in my case I have rounded speed to the km/h to create categorical distribution, for example 100 km/h represent one possible state of the model). My approximated emission matrix $B^{'}$ has same size as transition matrix (i.e. relate speeds from ASIM to speeds from FCD) and is defined as

$$B^{'} = \{b_j^{'}(y_t)\}, b_j^{'}(y_t) = P(Y_t = y_t | X_t = j),$$

where Y_t is observation variable (FCD). Values $b_j^{'}(y_t)$ based on measurement error of FCD, i.e. in physics, measurements tend to have Gaussian distribution of errors. Therefore, approximated emission matrix relates hidden states to emissions using Gaussian distribution of probability in each row, centered on the true value, so it is most probable that hidden state emit itself, less probable that it emits value slightly smaller or greater and so on. During my experiments, it has been found that Gaussian with slightly negative kurtosis performs the best in this task. I also define initial distribution state

$$\pi_i = P(X_1 = i).$$

which is based on probability on individual states of X_t.

The second step, training of the model, is performed by standard Baum-Welsh algorithm [20]. Both approximated matrices $A^{'}$ and $B^{'}$ and historical FCD speed time series Y(i.e. emissions) from the predicted segment are used as an input. This can be written as finding

$$\theta^* = \arg\max_{\theta} P(Y|\theta),$$

where Y is training observation sequence and $\theta = (A^{'}, B^{'}, \pi)$ is set approximated model parameters. As the result, I receive θ^* containing true transition and emission matrices (A and B) of the model. Note that trained model is only suitable for predicting the speed on FCD section on which it was trained.

In the next step, n emission sequences of certain length r are generated and connected to the last s values of the predicted FCD time series Yc creating extended emission sequence

$$e_i = [Yc_{t-s}, \ldots, Yc_t, \ldots, Yc_{t+r}],$$

where $i \in [1, n]$. This step is done to maintain the continuity between generated sequence and time series. Then each extended emission sequence probability $P(e_i|\theta^*)$ is calculated using Forward-backward algorithm [20]. The m most probable extended emission sequences are chosen and their weighted mean is calculated as:

$$p = \frac{1}{W} \sum_{i=1}^{m} w_i e_i,$$

where p is predicted sequence, w_i are individual weights and W is sum of all weights. This step is included to diminish probability of getting the false classification of predicted emission sequence. In my case, I was using linearly decreasing weights w. The result of this weighted mean p is my predicted emission sequence corresponding to FCD speed values. Pseudo code for the algorithm is written in Listing 3.

Algorithm 3 HMM prediction algorithm

procedure HMM PREDICT(X, Y, Yc, n, s, r)
  X: historical ASIM values
  Y: historical FCD values
  Yc: current FCD time series
  n: number of extended emissions
  s: number of last values of Yc
  r: length of the prediction
  θ=ApproximateTheta(X,Y); ▷ Compute approximate θ
  θ^*=BaumWelsh(θ,Y); ▷ Compute true θ^*
  for $i = 1 : n$ **do**
    e_i=$[Yc_{t-s}, \ldots, Yc_t, \ldots, Yc_{t+r}]$; ▷ Create extended emission e_i
    lik_i=Forward(e_i,θ^*); ▷ Calculate likelihood of e_i
  end for
  p=Predict(lik,e_i); ▷ Compute weighted mean of best predictions
end procedure

8.2.2 Experimental results

My algorithm was implemented in Matlab environment. From the performance perspective, only problematic part is training of HMM which can easily take tens of minutes. However, given enough historical data, it is not necessary to frequently update the model. The parallelized computation of prediction with trained model takes only tens of seconds.

The algorithm was tested on data coming from Czech D1 motorway from the period of 3.1.2014 to 30.4.2014. I have chosen section from 43rd to 47th km in the direction of Prague. This section contains one ASIM sensor, which was used for the calculation of the approximated transition matrix A'. I have used data from January to March for training and data from April for evaluation. Root Mean Square Error (mathematical description can be found in [2]) was chosen as a measure for quality of the prediction. Five minutes were chosen as a prediction step, due to the aggregation period of ASIM sensors, which is 5 minutes long. Each prediction produces six such steps, so the traffic speed during next 30 minutes is predicted.

The first experiment was performed on one such 30-minute prediction at randomly chosen time in April. I have done this prediction 100 times and compared the results in terms of accuracy. Graph of RMSE of individual predictions can be seen in Figure 23.

With mean RMSE equalling to 5.32 and none higher than 8.6, the algorithm performs comparably to other contemporary approaches. The best and the worst-case prediction determined by their RMSE are shown in the Figure 24. Prediction starts at a time window 20; previous time windows are the last values from the predicted FCD time series.

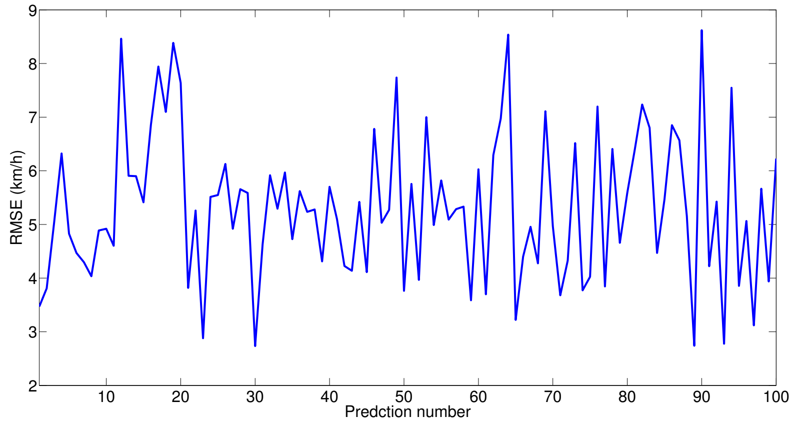

Figure 24: RMSE of individual predictions

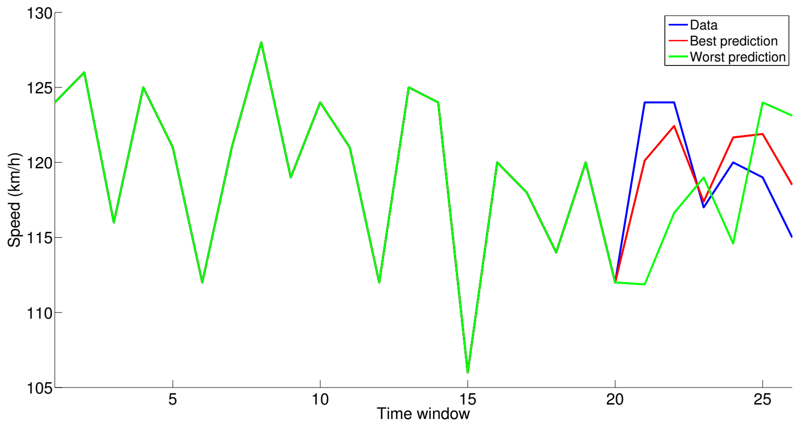

Figure 25: The best and the worst case speed prediction

In the best case, traffic speed is predicted almost perfectly both in terms of magnitude and trend. The worst-case scenario is still capable of capturing the trend but it is a bit delayed and magnitude is also distorted. However, it still very roughly describes the real state of the traffic speed.

In the second experiment, I have performed this prediction repeatedly for 25 times, each time moving the predicted window by the 30 minutes, creating the more than 12 hours of predicted speeds. Note that actual measured data were used to create extended emission sequences and not the ones predicted in previous step. Results can be seen in Figure 25.

It is evident that my prediction algorithm majority of time captures the trend but sometimes is off the magnitude. However, as it is shown in Figure 26, RMSE of individual predictions is still in reasonable limits.

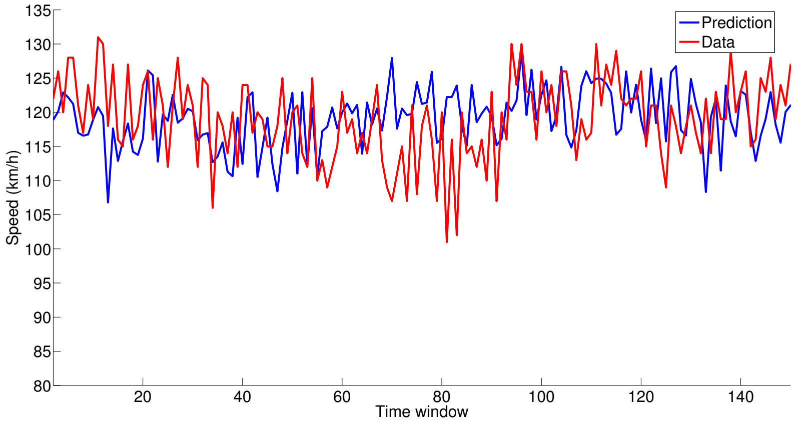

Figure 26: Predicted and real traffic speed during the 12 hours

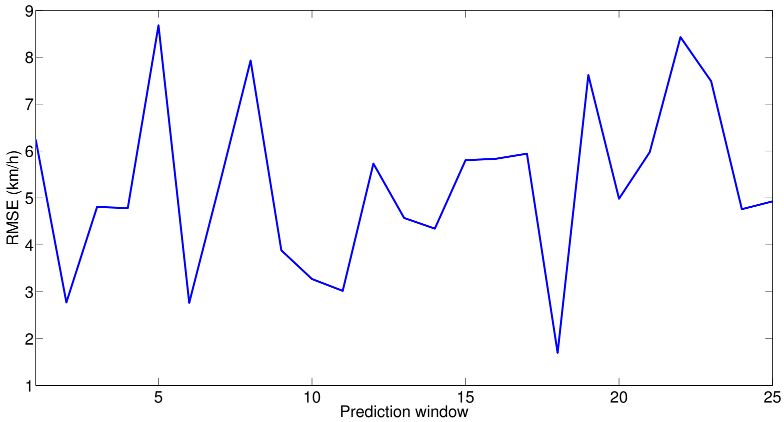

Figure 27: RMSE of individual predicted windows

8.3 Ensemble Kalman filter approach

8.3.1 Algorithm

Ensemble Kalman filter based algorithm is presented in this part. Parts of this approach were presented in [15] and [57]. Once again, this algorithm utilizes both historical and current data from both my data sources. Main difference between this algorithm and the other ones is, that this one predicts the speed on the entire motorway and not only parts of it. Also, this algorithm is not only based on the historical data but also on a physical model. As a standalone, physical model (in my case CTM-v) would be unable, due to the missing data regarding its boundary conditions, cope with this task. However, thanks to the smart combination of historical data and physical model it is possible to utilize it in the prediction. Moreover, thanks to the integration of CTM-v into Ensemble Kalman filter, efficient assimilation of current situation data is guaranteed. Ensemble Kalman filter was chosen, because the problem of the traffic modelling is generally considered to be non-linear and not suitable for general Kalman filter. Due to the assimilation step, this algorithm is more useful for relatively short predictions until next traffic state data are available (5 minutes in my case because of ASIMs). General progress of this algorithm can be summarized into following points (note that steps 3 and 4 are performed repetitively while the prediction is being performed and current data supplied):

1. Learn the boundary conditions from the historical data.

2. Initiate an ensemble.

3. Perform an ensemble of predictions based on the learned physical model.

4. Assimilate current state data by EnKF.

Let us discuss each of these steps in detail. In the first step, learning of the boundary conditions is performed. It can be likened to the inverse modelling task of finding a set of boundary conditions B, such that

$$B = \arg\min_{b} |m(x_0, b) - x|, \tag{22}$$

where b are possible boundary conditions, x are available data and function m represents my physical model (i.e. I am minimizing the difference between the physical model and data given the boundary conditions). Given that I need to define boundary condition for each ramp on the motorway and that this condition should change in time, this optimization is not a simple one. It can be expected, that minimizing Equation 22 would be very difficult due to the high number of local minimums. Solution of this problem was found in the field of evolutionary computations and is named Differential evolution. This optimization technique is well proven to handle such difficult optimization problems as for example is proven in [15]. Optimized variable b is a matrix where rows represent individual ramps and columns its progress in time (this forms my specimen). Crossbreeding between specimen is then performed between rows of the b. In an optimal situation, model should be optimized for a week-long period, as the week is larges strictly periodical time interval in traffic. These specimens are then inserted into a CTM-v model and their results are compared to the original data producing the measure of quality. Result of this optimizations is utilized in the step 3.

The second step is easy and straightforward. N members of ensemble are initialized by using free flow speed v_{ff} randomized by variance σ^2 obtained from the historical data for each cell:

$$v_i = v_{ff} + N(0, \sigma^2),$$

where v_i is initial speed in the cell.

The next phase is a prediction phase. This phase is based on CTM-v and is realized by a modification of Equation 17

$$v_i^{n+1} = v_i^n - \frac{\Delta t}{\Delta x} \left(g\left(v_i^n, v_{i+1}^n\right) - g\left(v_{i-1}^n, v_i^n\right) \pm B_i^n \right), \tag{23}$$

where B_i^n is control variable governing the boundary conditions and learned in previous step. Addition or deduction depend whether it is in-ramp (-) or off-ramp (+). Length of each cell x and time step t must be set to adhere to Courant-Friedrichs-Lewy conditions. This means that if I want to have a suitably small cell, each model step must be reasonably short and it is necessary to advance the model many times each prediction phase by calculating Equation 23 (for example, 5 minute prediction requires 300 prediction steps without any further data). This results in problem of finding balance between detail and accuracy and will be topic for the future research. Because this algorithm is using EnKF, this prediction is not only done once but for each N members of the ensemble. Resulting prediction from the entire ensemble can be obtained by calculating an interval of confidence for mean speed of the ensemble (as usually ensemble has more than a few members).

The last step of my algorithm is current state data assimilation. This step is performed once per minute or per 5 minutes depending whether ASIM sensor data are used. It is also quite straightforward as this step consist only of calculating ensemble prediction covariance P_t from Equation 7, calculating Kalman gain from Equation 8 and updating the ensemble by combining the new measurements with current ensemble by calculating the Equation 9. Measurements covariance R is calculated from historical data. Measurements used for this step come mostly from the FCD (TNC segments are divided into cells with the same measured value) and also can come from ASIM, where they are related to the cell with the gate. This provides us with more accurate measurement but also lengthens my prediction step. The entire algorithm can be summarized into pseudo code 4.

Algorithm 4 Ensemble Kalman filter prediction algorithm

procedure HMM PREDICT(X, Xc, N, R, $evol$)
    X: historical FCD and ASIM values
    Xc: current speed time series
    N: ensemble size
    R: measurements covariance
    $evol$: vector with settings for DE
    B=LearnDE(X,$evol$); ▷ Learning of boundary conditions by DE
    En=InitializeEnsemble(N,X); ▷ Ensemble initialization
    while x_c **do**
        $EnsemP$=PredictEnsemble($Ensem$,B); ▷ Prediction step
        $Ensem$=Assimilate(Xc,$EnsemP$,R); ▷ Assimilation of current measurements
    end while
end procedure

8.3.2 Experimental results

Experiments with this algorithm were performed on data from D1 (roughly from Humpolec to Brno) coming from 21.4.2014. Unfortunately, quantity of the data is insufficient to perform thorough testing and evaluation, however I do not possess larger data set at the moment. These results should be therefore considered as demonstration of usability. This problem would be solved soon with access to new datasets.

Experiment was performed only on FCD data, however, I kept 5-minute prediction step. I used 10 hours from 6:00 to 16:00 for learning and measurement covariance estimate. I used 50 generations of 75 individuals and set crossbreeding threshold to 0.7 and mutation constant to 1.2. After learning the model, I performed 20 prediction and assimilation steps. Spatial discretization for CTM-v model was chosen to be 100 m and temporal 1 second (by the CFL conditions). Results from prediction steps 6, 12 and 20 can be seen in Figures 28, 29, 30, 31, 32 and 33. These Figures show true traffic speed on entire section of the D1 motorway in the given step and prediction confidence interval.

RMSEs of these predictions are 8.65, 10.10 and 12.31 respectively. While these numbers may not seem to too high, few glances at the Figures 29, 31 and 33 show that results are roughly capable of capturing the magnitude of speed but rarely the trend. This fact seriously degrades value of the results. However, these problems are probably caused by small training set and may be solved by introduction of the new and larger data sets.

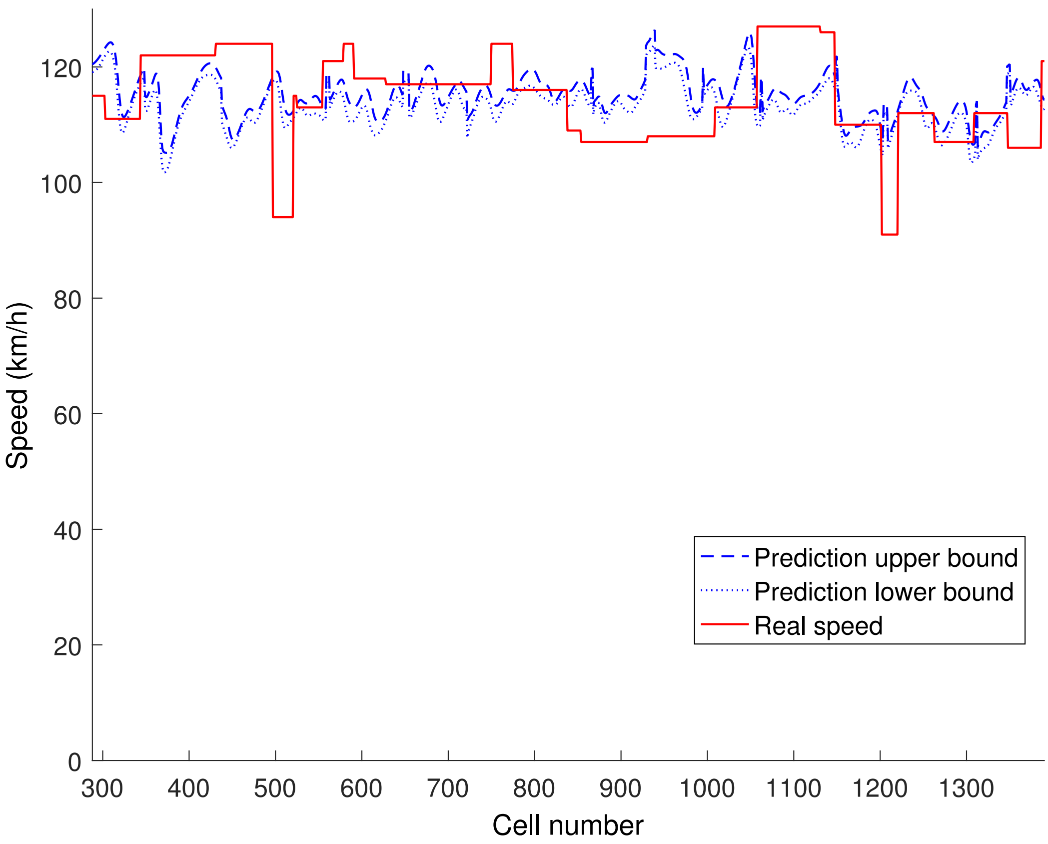

Figure 28: Traffic speed on segment of D1 after prediction step 6

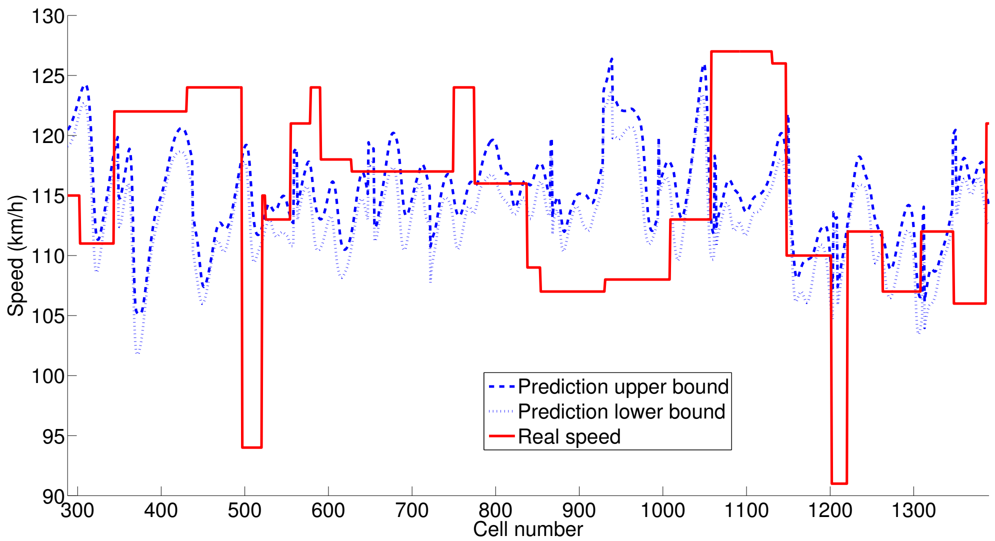

Figure 29: Traffic speed on segment of D1 after prediction step 6 (detail)

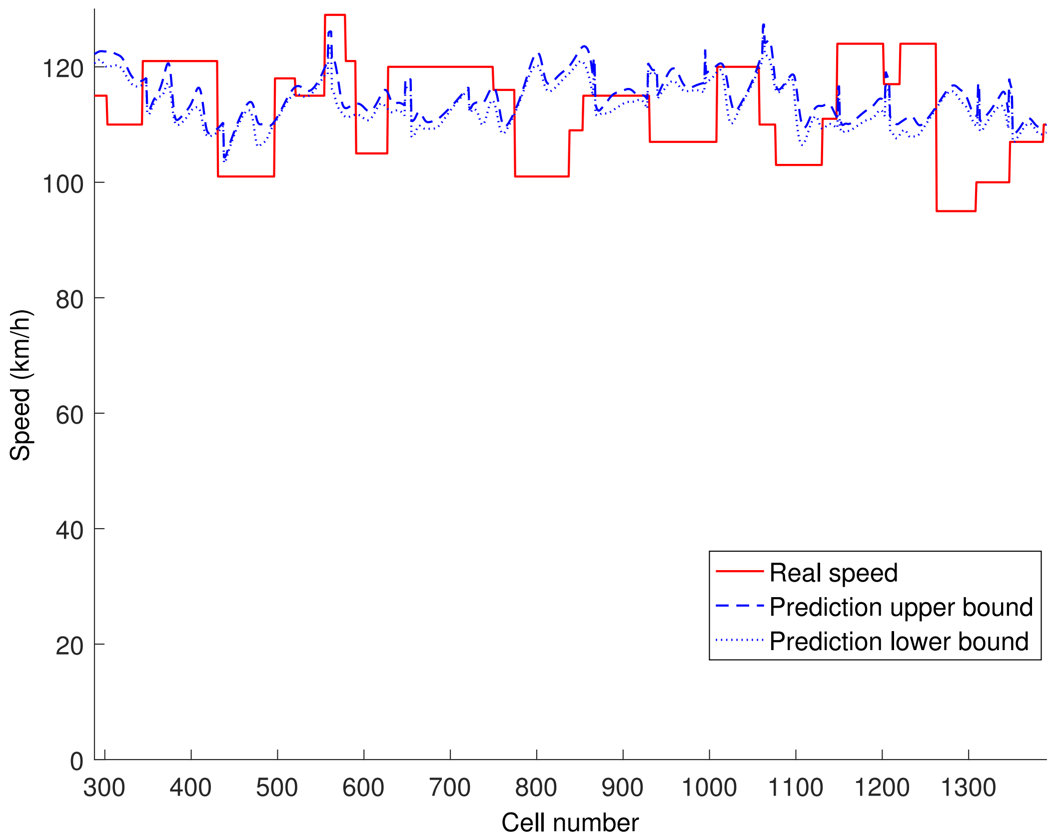

Figure 30: Traffic speed on segment of D1 after prediction step 12

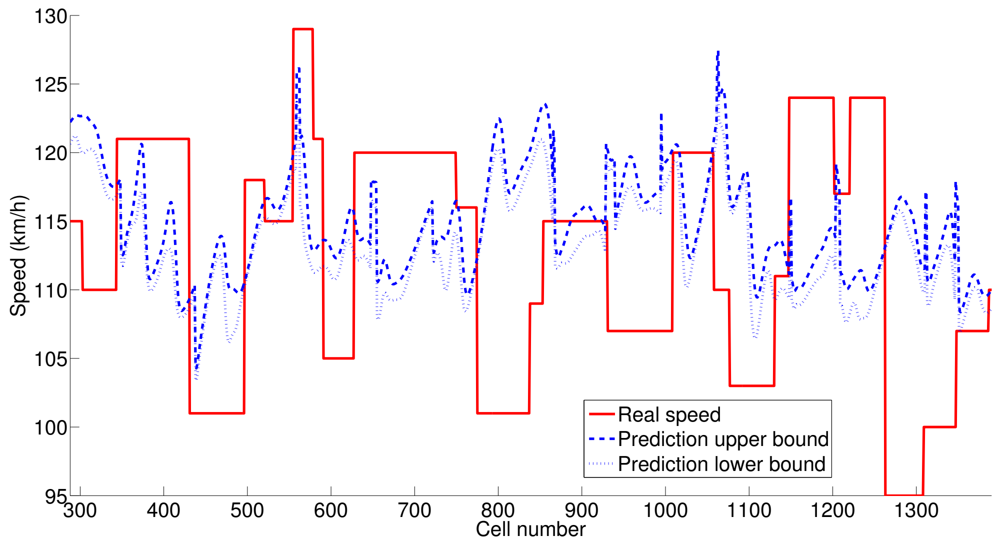

Figure 31: Traffic speed on segment of D1 after prediction step 12 (detail)

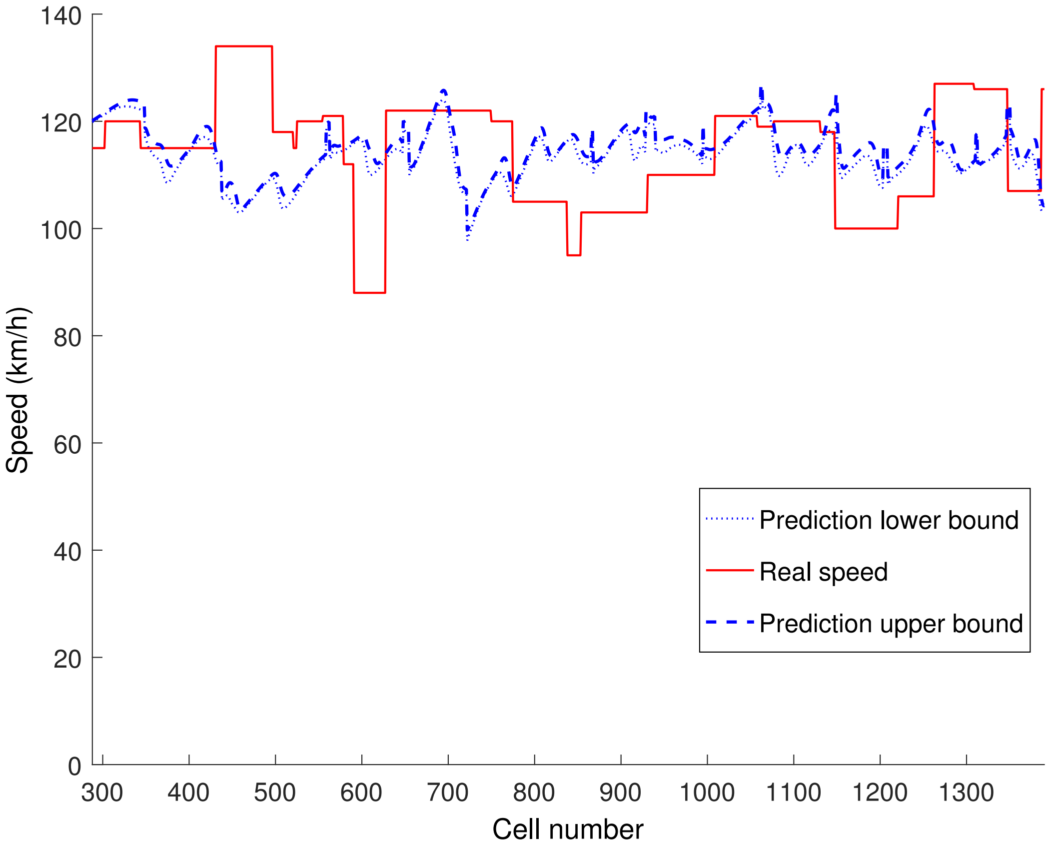

Figure 32: Traffic speed on segment of D1 after prediction step 20

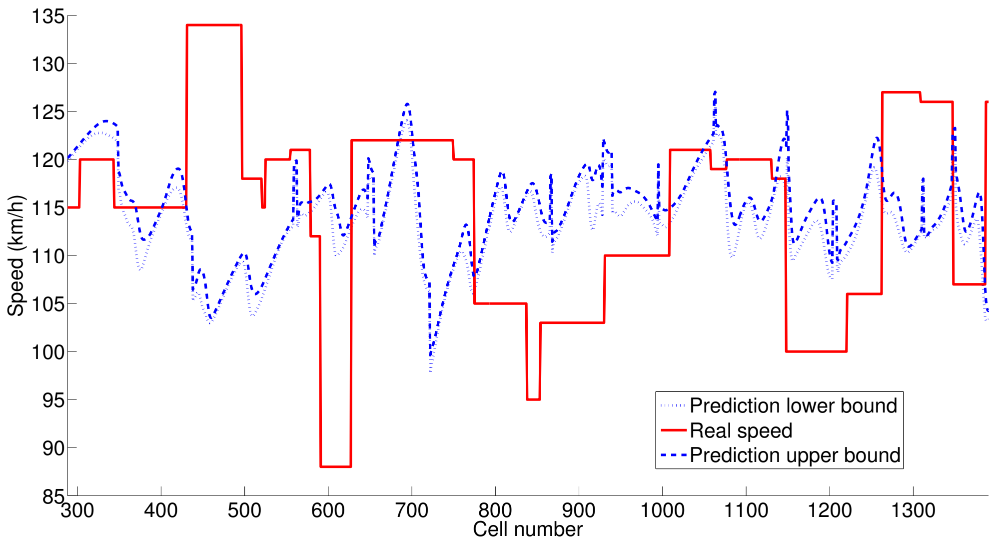

Figure 33: Traffic speed on segment of D1 after prediction step 20 (detail)

8.4 Markov chain model approach for traffic incident length prediction

8.4.1 Algorithm

This algorithm was published in[60] and [58]. As it has been mentioned in Section 6 and as it is true for all my algorithms, it is based on analysis of historical traffic data. These data are represented by speed time series for segments of the road (based on FCD). However, main aim of this algorithm is not predicting the speed of the traffic but predicting how long will the speed be under certain threshold during the traffic incident. This task is accomplished not through the direct prediction, but through the classification of traffic incidents and simulation based on this classification. For clearer understanding, progress of my general classification algorithm for traffic time series can be summarized into following steps:

1. Find traffic incidents in historical time series and perform a clustering on these incidents.
2. Find Markov chains for parametrization of these clusters.
3. Detect traffic incident in current traffic speed time series.
4. Classify these incidents by iterative Bayesian classification.
5. Based on this classification, simulate the traffic incident length.

Let us go through individual steps of the algorithm. In the first step, historical data are processed. This processing is divided into two parts. In the first part, traffic incidents in historical traffic speed time series are detected. This is done through dividing the time series into shorter ones, which have traffic speed under certain threshold. The parts with higher speeds are discarded. For example, I use only parts with values under 40 km per hour. This creates us number of short time series of various length representing traffic incidents. These historical incidents are then clustered to N classes of similar time series. This clustering is performed by Dynamic Time Warping approach [38]. This approach is ideal for this problem because it is capable of working with not only static patterns in the time series, but also their temporal distortions. At the end of this step, I have predefined number of classes (clusters) with each class containing a number of classified traffic incident time series.

Next step is cluster parameterization, which is done by Markov chains. Their main advantage is their simplicity and their low computational complexity. As Markov chains are finite state models, I have to transforms speed values from the time series to a number of states. Based on my experiments, I have decided to use one state for a span of ten km per hour (i.e. one state for speeds 0-9 km/h and so on). Initial state is the state that represents the value of speed just after the speed dropped under the critical level for incident detection. Absorbing state (i.e. state that I consider the end of speed time series for traffic incidents) is one state above the critical speed value. Transition matrices P_n are simply calculated from number of transitions between states in each on N clustered time series.

Step number three is quite straightforward. It is realized by simply detecting the drop of the traffic speed from FCD. When the speed is under certain threshold for five consecutive measurements (i.e. 5 minutes), classification step of the incident starts.

Next step of the algorithm is Bayesian classification of the new traffic incidents. I am using the classifier in form presented in Equation 21. It is even simplified by the fact, that I have just a single feature of the data (its time series) so I can drop the product from the equation. As a decision rule, I am using standard maximum posteriori probability rule. This produces equation

$$z = \operatorname*{arg\,max}_{n \in \{1,\dots,N\}} p(C_n) \cdot p(x_0, \dots, x_t | C_n). \tag{24}$$

A priory probability of the model $p(C_k)$ is calculated from the size of appropriate cluster, models of large clusters of historical time series are more probable than those of smaller clusters. More difficult part of the computation is the likelihood $p(x_0, \dots, x_t | C_k)$. It represents probability, that the model generates the data and in general is not exactly known and therefore, it must be simulated. I have used the Monte Carlo approach [63] by drawing M time series from the model C_n and comparing them to the data x. From a large number of these comparison, I can compute the likelihood of the data:

$$lik = \frac{1}{M} \sum_{i=1}^{M} \begin{cases} 1, \; if \; d(m_i, x_{\{1,\dots,t\}}) > tr \\ 0, \; else \end{cases} ,$$

where m_i are individual time series drawn from the model C_n and $tr \in < 0, 1 >$ is some threshold. This comparison $d(m_i, x_{\{1,\dots,t\}})$, however, creates another problem. These series will usually be of different length, but it does not mean that they are not similar. They can have only slightly different length but their shape is very similar. I can again solve this problem by application of dynamic time warping (DTW) for measuring time series similarity. If this similarity is greater than the threshold tr, I can consider that the data can be generated by the model. This entire algorithm for computation of a posteriori probability of a model can be described by following pseudo code:

This classification can be done in one of two ways: either on-line of off-line. On-line classification is performed each time I receive a new data regarding current traffic incident. It is exactly formulated by Equation 24 and is performed each time I receive new measurement from the traffic incident time series. This on-line classification serves as an input for the next step. Off-line classification is performed after the traffic incident ends. Purpose of this classification is to bring a new data into the dataset as these classified time series can be added to relevant cluster and parametrization of this cluster can be recalculated. In this way, the algorithm can adapt and iteratively improve its learning set.

The last step of the algorithm is current incident length prediction. This task is again performed by Monte Carlo simulation method. The M sample incident time series completions of the current time series $x_{1,\dots,t}$ are drawn from the classified model z and are joined to $x_{1,\dots,t}$.

Algorithm 5 Algorithm for computing a posterior probability of model C_k

procedure APOSTERIORPROBABILITY(C_k, $p(C_k)$, x, n, *treshold*)
  for $i = 1 : n$ **do**
    ts=simulate(C_k); ▷ Monte Carlo simulation
    sim=dtwsimilarity(*ts*,*x*); ▷ DTW similarity of data and simulated time series
    if *sim* > *treshold* **then**
      mc++;
    end if
  end for
  likelyhood=*mc*/*n*; ▷ Likelihood computation
  aposterior=*likelihood* · $p(C_k)$; ▷ A posterior probability computation
end procedure

Then their mean length representing prediction of duration of current incident p_l is calculated as

$$p_l = \frac{1}{M} \sum_{i=1}^{M} length([x_{1,\ldots,t} + m_i]),$$

where *length* denotes function calculating the length of time series. This p_l represent the predicted length of current incident and is iteratively updated every time I receive new data until the incident is over. Pseudo code for the entire algorithm is written in the Listing 6.

Algorithm 6 Markov chain incident duration prediction algorithm

procedure HMM PREDICT(X, Xc, N, M, *tr*, *tres*)
  X: historical FCD values
  Xc: current incident FCD time series
  N: number of classes for classification
  M: Monte Carlo iterations
  tr: similarity threshold
  tres: traffic incident speed threshold
  incidents=FindIncidents(X); ▷ Find traffic incidents in historical FCD time series
  clusters=ClusterDTW(*incidents*,N); ▷ Create N clusters from these incidents
  class=MarkovChain(*incidents*,N); ▷ Calculate Markov chain parametrization of *clusters*
  while $v_t <$ *tres* **do**
    z=BayesClassOn($Xc_1, \ldots, t$,*class*,*tres*,M); ▷ On-line Bayesian classification of $Xc_1, \ldots, t$
    p_l=Predict($Xc_1, \ldots, t$,z,M); ▷ Calculate prediction
    t++;
  end while
  class=BayesClassOff(Xc,*class*,*tres*,M); ▷ Off-line Bayesian classification of Xc and update of Markov chain parametrization
end procedure

8.4.2 Experimental results

My algorithm was tested on time series of speed from Czech Republic highways. Historical data used for training come from the April 2015 and testing traffic incidents come from May 2015. I have identified traffic incidents in these time series and extracted them. Standard DTW clustering performed on traffic incidents from April produced 25 clusters, which I labeled by IDs from 1 to 25(small clusters with just one or two incidents were not taken into account). These clusters have been transformed into 25 Markov chain models representing these clusters and numbered accordingly to their originating cluster. Then I have taken time series from May and clustered them by DTW to provide a benchmark (these clusters were again numbered for their identification). At first, I have tested the off-line classification. I took series from the three largest clusters created by this clustering and tried to classify them using my algorithm and Markov models obtained from April. It can be expected, that all-time series from each cluster should be classified to come from the same model. Results can be seen in Table 3.

ID of model representing classified series	ID of originating cluster
1	1
3	1
1	1
1	1
24	1
1	1
1	10
10	10
10	10
10	10
5	5
7	5
5	5

Table 3: Comparison of originating cluster and results of classification

As it is evident from the Table 3, my algorithm can successfully classify most of the time series but not all of them. While the success rate suggests that algorithm is functioning quite well, it is also showing some flaws. Let us analyse the first six time series, which should all belong to the same model, to see the reason behind the misclassified series. In Figure 27 you can see all six time series. Correctly classified ones are blue, the one from the second row is red and the one from the fifth row is green.

The red time series is noticeably shorter than the other ones. It may have been on a border of the cluster and therefore it is possible that some other model describes it better. The green one is, however, very similar to the other ones. Problem here lies with its a posterior probability which is shown in Figure 34 (axis x shows ID number of model and axis y its a posterior probability).

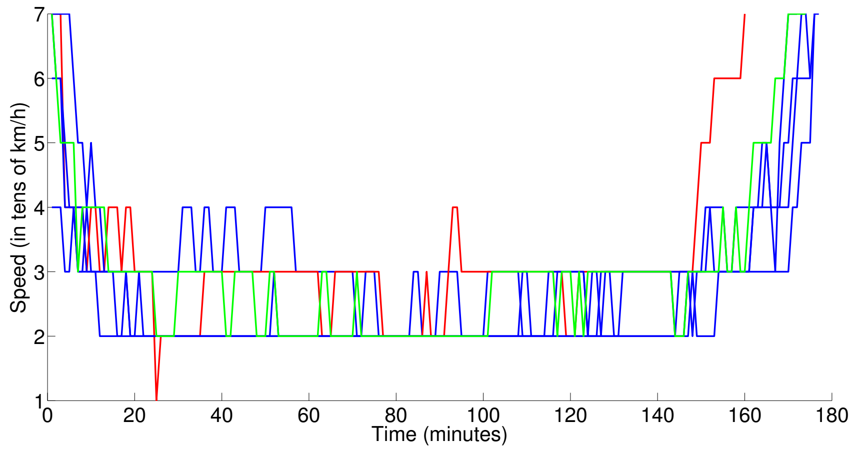

Figure 34: Time series of speed from cluster ID 1

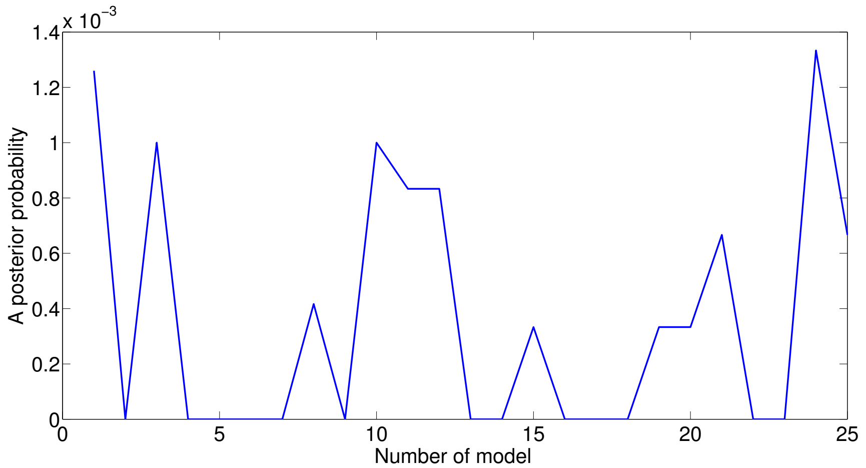

Figure 35: A posterior probability distribution of time series from the fifth row of Table 3

Its MAP peaks at model 24, but it is only marginally higher that a posterior probability at model 1. This time, it is clearly misclassification, as series from cluster 24 are shorter. As likelihood values are very small, it is possible to misclassify the time series because likelihood tended to be in favour of wrong model. With these minor exceptions, the algorithm for off-line classification performs suitably well.

On-line iterative classification was tested on two traffic incidents from May 2015 (the first is 47 and the second is 44 minutes long). Starting ten minutes after the start of incident, current state was classified each minute and prediction about it length simulated. Results can be seen in Figures 36 and 37.

While results from the both figures seem reasonably good with occasional misclassification, their RMSEs are quite high at 60.36 in case of Figure 36 and 60.84 in case of Figure 37. However, this is due to problematic distribution of errors in individual classifications as most of the time,

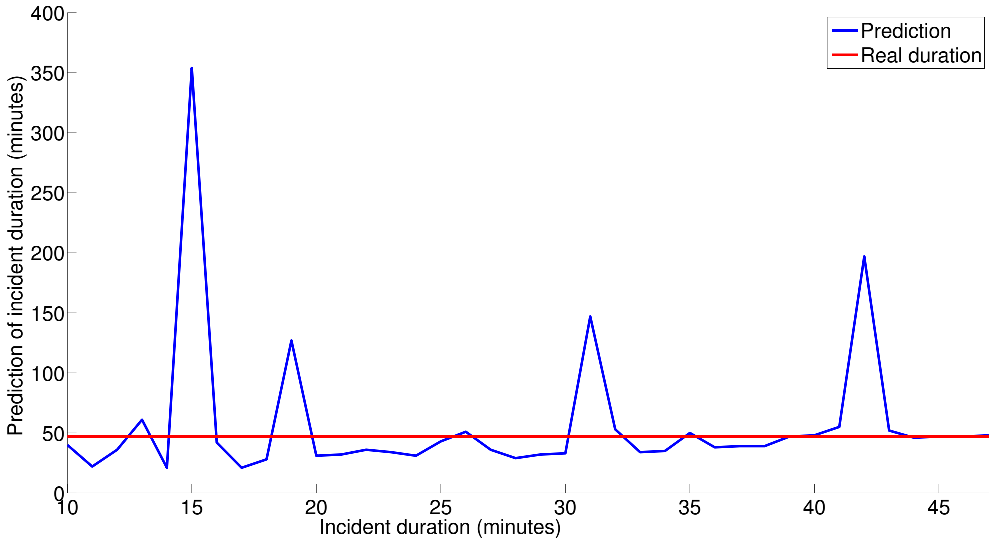

Figure 36: Prediction of length of the first incident

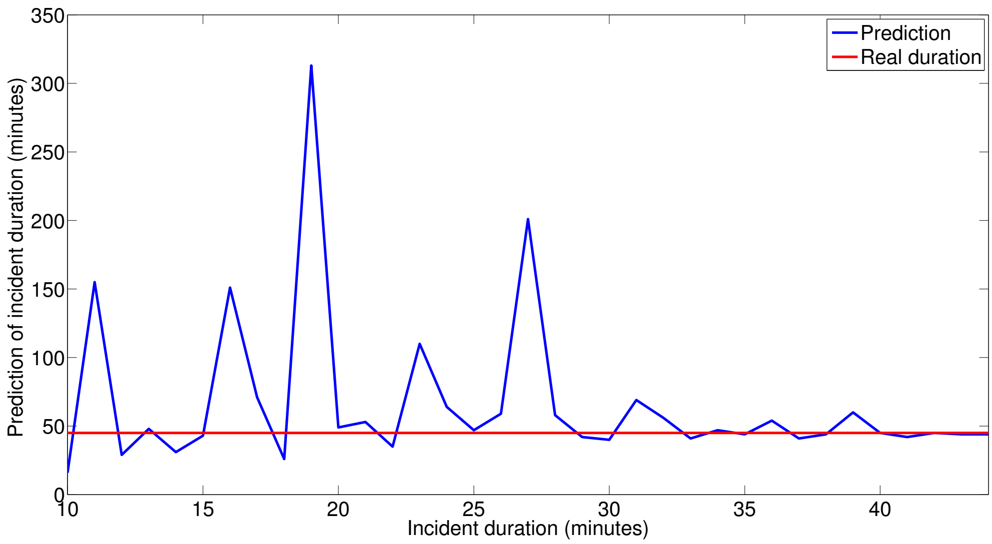

Figure 37: Prediction of length of the first incident

error is very reasonable, but there are some outliers as can be seen in Figures 38 and 38. Solution of this problem is discussed in the Conclusion. Except for these small problems, algorithm for on-line traffic incident length prediction perform very well.

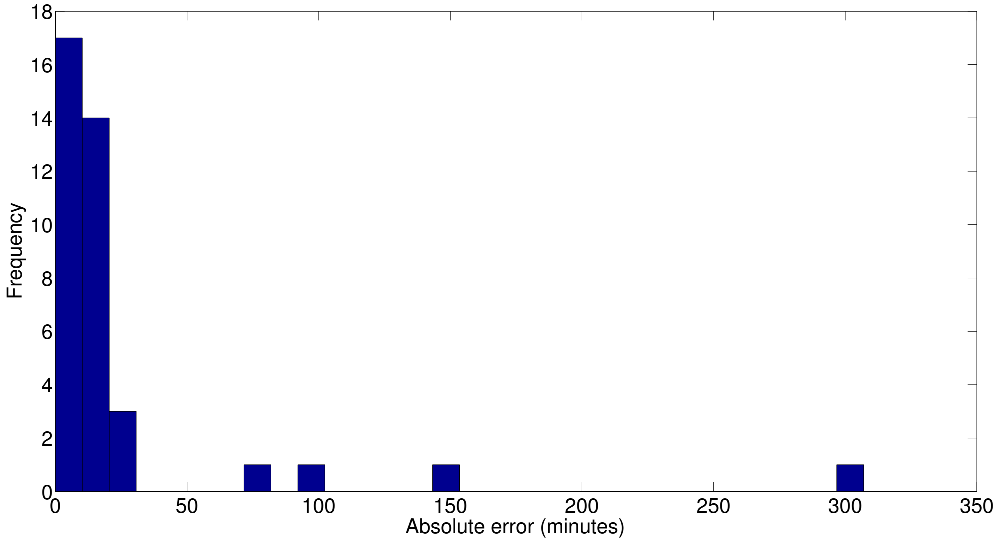

Figure 38: Distribution of absolute error of individual predictions from Figure 36

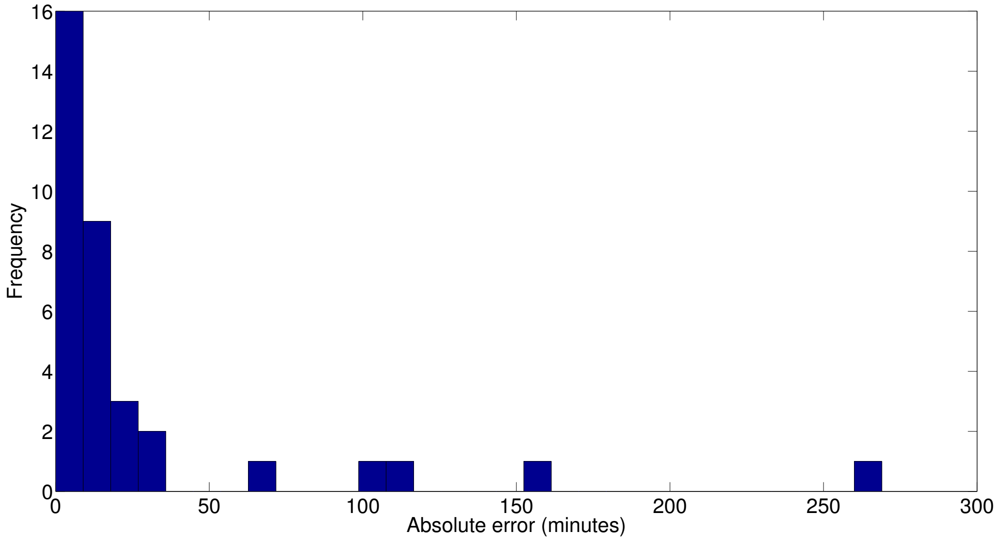

Figure 39: Distribution of absolute error of individual predictions from Figure 37

9 Conclusion

9.1 Discussion of results

Results of the developed algorithms are discussed, compared to each other and compared to the results from contemporary works in this part. Algorithms were tested on various real traffic data from D1 motorway. Four algorithms were developed, three for the prediction of traffic speed and one for the prediction of traffic incident duration.

Comparing results from the three traffic speed prediction algorithms in Table 4, it can be easily identified that the best results (measured by their RMSE) were given by HMM algorithm, followed by the DBN and EnKF algorithms.

Method	HMM algorithm	DBN algorithm	EnKF algorithm
Average RMSE	5.32	8.91	10.35

Table 4: Comparison of RMSE of implemented algorithms

Moreover as it is evident from the Figures 25, 26, 29,31, 33, 21, 22 and 23, algorithms perform similarly well in terms of catching the trend of the traffic speed. While, even in case of EnKF approach, RMSE stays within reasonable limits of 15% of mean, EnKF and DBN algorithms have some reserves in this comparison. These results are caused by the nature of the data and algorithms. Due to the legal and licensing issues, my testing data were usually of limited size. This fact proved to be comparative advantage for HMM algorithm because it is strictly localized and therefore requiring less data for proper learning. Both other algorithms were developed for the prediction of speed for larger parts of the road. This is mixed blessing in my case, because while these models may exploit dependencies which cannot be utilized by HMM model, they also contain much more parameters to be learned and need much larger learning sets. When improperly learned, these models produce lower quality results. This problem, however, will be addressed in near future, because traffic data fusion engine is currently being implemented for the ANTAREX project. This engine will provide much higher and broader volume of traffic data regarding Czech Republic road network and will serve as excellent basis for further testing and improvement of developed algorithms.

For the comparison to the contemporary works, I have chosen articles written by Coric et. al. [71], Jiang et. al. [35], Castillo et. al. [8] and Xie et. al [80]. Coric et. al. [71] used Mean Absolute Error (MAE) which are generally smaller than RMSE for equivalent data. These works were chosen because they present similar solutions and also have comparable results. Resulting errors of their algorithms are shown in Table 5.

Author	Coric et. al. [71]	Jiang et. al. [35]	Castillo et. al. [8]	Xie et. al [80]
Average RMSE	6.2 (MAE)	9	8.7	9.1

Table 5: Comparison of errors (RMSE or MAE) of contemporary works

From the table, it is evident that my algorithms have similar error results, even with small data sets. HMM algorithm even performs better than the other ones. Also, advantage of my algorithms is, that they were tested on real data with all of their problems and uncertainties. This is for example not true in case of Jiang et. al. [35] who used artificial data from SUMO traffic simulator.

In case of my traffic incident length prediction algorithm, I can only compare it to the results from the other authors. I have chosen work of Ruimin et. al. [42] who tries to solve very similar problem. Their predictions for incidents longer that 10 minutes achieved RMSE of 31.25 minutes. This number is much better than my average RMSE of 60.55. This is, however, not due to the fact that their model produces better estimates all the time, but due to some outliers in my prediction. As it is evident from the Figures 38 and 39, most of the time my prediction error would be very similar or smaller than theirs. Solving how to detect and filter these outlying misclassification or simulations will be the topic for my future research. Without this problem, my algorithm would have performed similarly or slightly better than contemporary works.

9.2 Conclusive remarks and future work

In this part, I would like to summarize scientific and application achievements of this work and outline some topics for future work and development of proposed methods.

From the scientific perspective, this work contains many interesting and original solutions. While proposed algorithms are generally based on existing methods, some subtasks required development of specific approaches. In case of DBN algorithm, scientifically most interesting achievement is the problem formulation and network structure which effectively utilizes both kinds of data sources and overcomes their lack of. In case of HMM algorithm, most important scientific contribution is the way in which are predictions created and evaluated and effective utilization of data sources. While both Naive Bayesian classification and Markov chains were already utilized in the area of traffic, developed algorithm is novel way of integration of both of these methods for off-line and on-line classification and prediction. The most important innovation of EnKF algorithm is utilization of differential evolution for inverse modelling of boundary conditions. Also, statistical analysis of relation of static and FCD data sources for Czech Republic is scientifically interesting because there was no such study before its publication.

Perhaps more important achievement of this work is from the perspective of its application. Experimental results of all proposed algorithms indicate their usefulness in real intelligent traffic systems. This fact also implies that all goals of this thesis stated in Section 3 were met. All three algorithms for traffic speed prediction can be, after some refinements and further tests, utilized for real prediction and they can be utilized in parallel. This deployment may prove to be beneficial as each of these algorithms has different resolution (from predicting the speed of single road segment to predicting the state of large parts of entire motorway). When the traffic incident happens, my Markov chain based algorithm can be utilized for predicting the length of this incident. Therefore, these algorithms offer complex solution which can be utilized in Digital Model of Mobility in RODOS system. Their usefulness is, however, not limited to the RODOS project as they will be utilized also in the ANTAREX project. Their task within this project will be to improve dynamic traffic routing by traffic speed predictions. Also, this project will provide much broader data coverage of the traffic so these algorithms can be extended for the prediction of other traffic quantities, like traffic density.

From the perspective of future work, developed algorithms can still utilize some improvements to overcome their minor issues (other than enlarging the learning data sets). In case of DBN algorithm, more efficient optimization in EM algorithm should be utilized. This optimization should be done through efficient parallelization on HPC. Parallelization of the algorithm will help to improve smaller accuracy of the model in case of presence of the hidden variable. Also experiments with different network structure will be performed. Similar improvement can be made for HMM algorithm. It will also utilize improved learning as current implementation can only handle few months of historical data. It is probably also task for HPC computing. Most critical issue of Markov chain algorithm is to find solution how to handle occasional misclassifications during on-line classification. Solution to this problem will significantly improve accuracy of the results. It will be a topic for my next research. When trained on the suitably large data set, EnKF algorithm will be improved by thorough calibration. The purpose of this calibration is to find the best spatial and temporal discretization and also the optimal prediction window. All of the algorithms will also be extended for prediction of other traffic quantities, namely traffic density and intensity. Also, combination of these attributes can be utilized in our models to provide higher level of prediction accuracy. This combination could improve the prediction because of known interaction of these attributes, which simplifies identification of different traffic situations and reduces uncertainty.

References

[1] J. Adachi A. Kinoshita, A. Takasu. Traffic incident detection using probabilistic topic model. In *Workshop Proceedings of the EDBT/ICDT 2014 Joint Conference*, 2014.

[2] J. S. Armstrong and C. Fred. Error measures for generalizing about forecasting methods: Empirical comparisons. *International Journal of Forecasting*, 8(1), 1992.

[3] M. T. Asif, J. Dauwels, C. Y. Goh, A. Oran, E. Fath, M. Xu, M. M. Dhanya, N. Mitrovic, and P. Jaillet. Unsupervised learning based performance analysis of v-support vector regression for speed prediction of a large road network. In *Proceedings of 15th International IEEE Conference on Intelligent Transportation Systems*, 2012.

[4] X. Boyen and D.Koller. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98),*, pages 33–42, 1998.

[5] S. C. Calvert, T. Henk, M. Snelder, and S. P. Hoogendoorn. Application of advanced sampling for efficient probabilistic traffic modelling. *Transportation Research Part C: Emerging Technologies*, 49, 2014.

[6] O. Cappe, E. Moulines, and T. Ryden. *Inference in Hidden Markov Models*. Springer, 2007.

[7] E. Castillo. Predicting traffic flow using bayesian networks. *Transportation Research Part B*, 42:482–509, 2008.

[8] E. Castillo. Stochastic demand dynamic traffic models using generalized beta-gaussian bayesian networks. *IEEE Transactions on Intelligent Transportation Systems*, 13, 2012.

[9] J. Laurie Snell Charles M. Grinstead. *Introduction to Probability*. American Mathematical Society, 2003.

[10] I. Zelinka; S. Celikovsky; H. Richter; G. Chen., editor. *Evolutionary Algorithms and Chaotic Systems*. Springer-Verlag Berlin Heidelberg, 2010.

[11] G. Costesequea and J. P. Lebacque. A variational formulation for higher order macroscopic traffic flow models: Numerical investigation. *Transportation Research Part B: Methodological*, 70, 2014.

[12] T. Kocyan D. Fedorcak. viarodos: Monitoring and visualisation of current trafficc situation on highways. In *Proceedings of the 13th IFIP TC8 International Conference on Computer Information Systems and Industrial Management, CISIM 2014*, 2014.

[13] C. de Fabritiis, R. Ragona, and G. Valenti. Traffic estimation and prediction based on real time floating car data. In *Proceedings of 11th International IEEE Conference on Intelligent Transportation Systems, 2008. ITSC 2008.*, pages 197–203, 2008.

[14] A. Doniec, R. Mandiau, S. Piechowiak, and S. Espie. A behavioral multi-agent model for road traffic simulation. *Engineering Applications of Artificial Intelligence*, 21:1443–1454, 2008.

[15] P. Dubec, J. Plucar, and L. Rapant. Use of the bio-inspired algorithms to find global minimum in force directed layout algorithms. In *Proceedings of the 6th International Conference on Multimedia Communications, Services and Security*, volume 368 of *Communications in Computer and Information Science*, pages 194–203. Springer Verlag, 2013.

[16] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[17] G. Welch G. Bishop. An introduction to the kalman filter. *SIGGRAPH*, 8, 2001.

[18] D. C. Gazis. *Traffic theory.* Kluwer Academic Publishers, 2002.

[19] L. Georgescu, D. Zeitler, and C. R. Standridge. Intelligent transportation system real time traffic speed prediction with minimal data. *Journal of Industrial Engineering and Management*, 5(2), 2012.

[20] Z. Ghahramani. An introduction to hidden markov models and bayesian networks. *International Journal of Pattern Recognition and Artificial Intelligence*, 15:9–42, 2001.

[21] B. Ghosh, B. Basu, and M. O'Mahony. Multivariate short-term traffic flow forecasting using time-series analysis. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):246–254, 2009.

[22] G. Gopi, J. Dauwels, M. Asif, S. Ashwin, N. Mitrovic, U. Rasheed, and P. Jaillet. Bayesian support vector regression for traffic speed prediction with error bars. In *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems*, 2013.

[23] C.W.J. Granger. Some recent development in a concept of causality. *Journal of Econometrics*, 39(1-2):199–211, 1988.

[24] A. Graser, M. Dragaschnig, W.Ponweiser, H. Koller, M. Marcinek, and P. Widhalm. Fcd in the real world – system capabilities and applications. In *Proceedings of 19th ITS World Congress, Vienna, Austria*, page 7, 2012.

[25] B. D. Greenshields. A study of traffic capacity. *Highway Research Board*, 14, 1935.

[26] J. Guo, W. Huang, and B. M. Williams. Adaptive kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transportation Research Part C*, 43:50–64, 2014.

[27] R. Herring, A. Hofleitner, P. Abbeel, and A. Bayen. Estimating arterial traffic conditions using sparse probe data. In *Proceedings of Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference*, pages 929–936, 2010.

[28] A. Hofleitner. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network. *IEEE Transactions on Intelligent Transportation Systems*, 13:1679–1693, 2012.

[29] P. K. Hoong, I. K. T. Tan, O. K. Chien, and C. Y. Ting. Road traffic prediction using bayesian networks. In *Proceedings of Wireless Communications and Applications (ICWCA 2012), IET International Conference*, pages 1–5, 2012.

[30] M.-L. Huang. Intersection traffic flow forecasting based on v-gsvr with a new hybrid evolutionary algorithm. *Neurocomputing*, 147:343–349, 2015.

[31] IITB. `https://www.civil.iitb.ac.in/tvm/1111_nptel/524_AutoMer/plain/plain.html`. 21.11.2016.

[32] IITB. `https://www.civil.iitb.ac.in/tvm/1111_nptel/512_FundRel/plain/plain.html`. 21.11.2016.

[33] Y. Ji, W. Daamen, X. Zhang, and L. Sun. Traffic incident recovery time prediction model based on cell transmission model. In *Proceedings of the 12th International IEEE Conference on Intelligent Transportation Systems*, 2009.

[34] Jiang. Short-term speed prediction using remote microwave sensor data: Machine learning versus statistical model. *Mathematical Problems in Engineering*, 2016, 2016.

[35] B. Jiang and Y. Fei. Traffic and vehicle speed prediction with neural network and hidden markov model in vehicular networks. In *Proceedings of 2015 IEEE Intelligent Vehicles Symposium*, 2015.

[36] M. Jones, Y. Geng, D. Nikovski, and T. Hirata. Predicting link travel times from floating car data. In *Proceedings of International IEEE Conference on Intelligent Transport Systems (ITSC)*, 2013.

[37] R. Khan, B. Landfeldt, and A. Dhamdher. Predicting travel times in dense and highly varying road traffic networks using starima models. Technical report, School of Information Technologies, The University of Sydney and National ICT Australia, 2012.

[38] T. Kocyan, J. Martinovic, and M. Podhoranyi. Searching and indexing distorted data collections. In *Proceedings of 26th European Modeling and Simulation Symposium, EMSS 2014*, 2014.

[39] G. Kuhns, R. Ebendt, P. Wagner, A. Sohr, and E. Brockfeld. Self evaluation of floating car data based on travel times from actual vehicle trajectories. In *Proceedings of IEEE Forum on Integrated and Sustainable Transportation Systems*, 2011.

[40] J. P. Lebacque. The godunov scheme and what it means for first order traffic flow models. In *Proceedings of 13th International Symposium on Transportation and Traffic Theory*, 1996.

[41] M. Li, Y. Zhang, and W. Wang. Analysis of congestion points based on probe car data. In *Proceedings of International IEEE Conference on Intelligent Transportation Systems, ITSC '09*, pages 1–5, 2009.

[42] R. Lia, F. C. Pereirab, and M. E. Ben-Akiva. Competing risks mixture model for traffic incident duration prediction. *Accident Analysis & Prevention*, 75, 2015.

[43] M. Lighthill and G. Whitham. On kinematic waves. ii. a theory of traffic flow on long crowded roads. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 229, no. 1178*, 1955.

[44] Y. Ma, H. J. van Zuylen, and J. van Dalen. Freight origin-destination matrix estimation based on multiple data sources: Methodological study. In *Proceedings of TRB 2012 Annual Meeting*, 2012.

[45] D. S. McLeod, L. Elefteriadou, and L. Jin. Travel time reliability as a performance measure: Applying florida's predictive model on the state's freeway system. In *Proceedings of TRB 2012 Annual Meeting*, 2012.

[46] Microsoft. http://social.microsoft.com/Forums/getfile/103119. 21.11.2016.

[47] V. Mihajlovic and M. Petkovic. *Dynamic Bayesian Networks: A State of the Art*. Report from Faculty of Electrical Engineering, Mathematics and Computer Science of University of Twente, 2001.

[48] T. Mitchell. *Machine Learning*. McGraw-Hill, Inc., 1997.

[49] F. V. Jensen; T. D. Nielsen. *Bayesian Networks and Decision Graphs (second edition)*. Springer Verlag, 2007.

[50] K. Parry and M. L. Hazelton. Bayesian inference for day-to-day dynamic traffic models. *Transportation Research Part B*, 50:104–115, 2013.

[51] G.H. Peng, X.H. Cai, C.Q. Liu, and B.F. Cao. A new lattice model of traffic flow with the consideration of the driver's forecast effects. *Physics Letters A*, 375:2153–2157, 2011.

[52] K. Price. Genetic annealing. *Dr. Dobb's Journal*, pages 127–132, 1994.

[53] K. Price. An introduction to differential evolution. *New Ideas in Optimization*, pages 79–108, 1999.

[54] K. Price and R. Storn. Differential evolution - a simple evolutionary strategy for fast optimization. *Dr. Dobb's Journal*, 264:12–24, 1997.

[55] Profingenerstroy. http://www.profingenerstroy.ru/production/. 21.11.2016.

[56] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE 77 (2)*, 1989.

[57] L. Rapant. Traffic flow modeling based on sparse data. In *Proceedings of WOFEX 2014*, pages 526–531. VSB-TUO, 2014.

[58] L. Rapant. Markov chain model approach for traffic incident length prediction. In *Proceedings of International Conference on Intelligent Traffic and Transportation 2017*, 2017. (Accepted).

[59] L. Rapant, T. Martinovic, K. Slaninova, and J. Martinovic. Traffic speed prediction using probabilistic graphical models. In *Proceedings of the 3rd International Conference on Traffic and Transport Engineering (ICTTE)*, pages 941–948. Scientific Research Center LTD Belgrade, 2016.

[60] L. Rapant, K. Slaninova, and J. Martinovic. Traffic incident classification using bayesian approach and markov chains. In *Proceedings of the 13th International Symposium on Operational Research*, pages 143–148. Slovenian Society Informatika, 2015.

[61] L. Rapant, K. Slaninova, J. Martinovic, and T. Martinovic. Traffic speed prediction using hidden markov models for czech republic highways. In *Proceedings of the 10th KES International Conference on Agent and Multi-Agent Systems: Technology and Applications*, volume 58 of *Smart Innovation, Systems and Technologies*, pages 187–196. Springer Science and Business Media Deutschland GmbH, 2016.

[62] L. Rapant, K. Slaninova, J. Martinovic, M. Scerba, and M. Hajek. Comparison of asim traffic profile detectors and floating car data during traffic incidents. In *Proceedings of 14th IFIP TC 8 International Conference on Computer Information Systems and Industrial Management*, volume 9339 of *Lecture Notes in Computer Science*, pages 120–131. Springer Verlags, 2015.

[63] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method.* Wiley-Interscience, 2007.

[64] R.-P. Schafer, K.-U. Thiessenhusen, and P. Wagner. A traffic information system by means of real-time floating-car data. In *Proceedings of ITS World Congress 2002, Chicago, USA*, 2002.

[65] S. Shiliang, Z. Changshui, and Z. Yi. Traffic flow forecasting using a spatio-temporal bayesian network predictor. *Artificial Neural Networks: Formal Models and Their Applications - ICANN 2005, Lecture Notes in Computer Science*, 3697:273–278, 2005.

[66] J. Lei; P. Bickel; C.M. Snyder. Comparison of ensemble kalman filters under non-gaussianity. *Monthly Weather Review*, 138:1293–1306, 2010.

[67] S. Sun, C. Zhang, and G. Yu. A bayesian network approach to traffic flow forecasting. *IEEE Transactions on Intelligent Transportation Systems*, 7:124–132, 2006.

[68] T.Q. Tan, H.J. Huang, and H.Y. Shang. A new macro model for traffic flow with the consideration of the driver's forecast effect. *Physics Letters A*, 374:1668–1672, 2010.

[69] T.-Q. Tang, L. Caccetta, Y.-H. Wu, H.-J Huan, and X.-B. Yang. A macro model for traffic flow on road networks with varying road conditions. *Journal of Advanced Transportation*, 48:304–317, 2014.

[70] T. Q. Tang, J. G. Li, H. J. Huang, and X. B. Yang. A car-following model with real-time road conditions and numerical tests. *Measurement Volume*, 48, 2014.

[71] S. Vucetic V. Coric, N. Djuric. Traffic state estimation from aggregated measurements with signal reconstruction techniques. *Transportation Research Record: Journal of the Transportation Research Board*, 2315, 2012.

[72] E. I. Vlahogianni. Enhancing predictions in signalized arterials with information on short-term traffic flow dynamics. *Journal of Intelligent Transportation Systems*, 13(2):73–84, 2009.

[73] J. Wang, W. Deng, and Y. Guo. New bayesian combination method for short-term traffic flow forecasting. *Transportation Research Part C*, 43:79–94, 2014.

[74] X. Wang, L. Peng, T. Chi, M. Li, X. Yao, and J. Shao. A hidden markov model for urban-scale traffic estimation using floating car data. *PLoS ONE*, 10, 2015.

[75] Y. Wang and M. Papageorgiou. Real-time freeway traffic state estimation based on extended kalman filter: a general approach. *Transportation Research Part B*, 39:141–167, 2005.

[76] D. Work, O. Tossavainen, S. Blandin, A. Bayen, T. Iwuchukwu, and K. Tracton. An ensemble kalman filtering approach to highway traffic estimation using gps enabled mobile devices. In *Proceedings of the 47th IEEE Conference on Decision and Control*, 2008.

[77] D. B. Work. A traffic model for velocity data assimilation. *Applied Mathematics eXpress*, 2010(1), 2010.

[78] Y. Wu, F. Chen, C. Lu, and B. Smith. Traffic flow prediction for urban network using spatio-temporal random effects model. In *Proceedings of the 91st Annual Meeting of the Transportation Research Board*, 2011.

[79] K. Liu X. Fei, C-C. Lu. A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. *Transportation Research Part C*, 19:1306–1318, 2011.

[80] Y. Xie, Y. Zhang, and Z. Ye. Short-term traffic volume forecasting using kalman filter with discrete wavelet decomposition. *Computer-Aided Civil and Infrastructure Engineering*, 22:326–334, 2007.

[81] K. Yan. Prediction of short-term traffic variables using intelligent swarm-based neural networks. *IEEE Transactions on Control Systems Technology*, 21:263–274, 2013.

[82] J. W. Yoon and T. T. Tchrakian. Bayesian implementation of a lagrangian macroscopic traffic flow model. In *Proceedings of 21st International Conference on Pattern Recognition (ICPR 2012)*, 2012.

[83] W. Zeng, Z. He, R. Lu, Zhuang, Lijiang, and X. Xia. Freeway segment speed estimation model based on distribution features of floating-car data. In *Proceedings of TRB 2012 Annual Meeting*, 2012.

[84] J. Zheng. Road travel time estimation with gps floating car data. In *Proceedings of Stander Symposium Posters. Book 191*, 2012.

[85] Z. Zhu, B. Peng, C. Xiong, and L. Zhang. Short-term traffic flow prediction with linear conditional gaussian bayesian network. *J. Adv. Transp*, 2016.

A Author's publications

A.1 Related to the Thesis

A.1.1 Proceedings articles (indexed)

- L. Rapant, K. Slaninova, J. Martinovic, M. Scerba, and M. Hajek. Comparison of asim traffic profile detectors and floating car data during traffic incidents. In *Proceedings of 14th IFIP TC 8 International Conference on Computer Information Systems and Industrial Management*, volume 9339 of *Lecture Notes in Computer Science*, pages 120–131. Springer Verlag, 2015. (SCOPUS SJR 0.315, WoS, major contribution)

- P. Dubec, J. Plucar, and L. Rapant. Use of the bio-inspired algorithms to find global minimum in force directed layout algorithms. In *Proceedings of the 6th International Conference on Multimedia Communications, Services and Security*, volume 368 of *Communications in Computer and Information Science*, pages 194–203. Springer Verlag, 2013. (SCOPUS SJR 0.162, WoS, major contribution)

- L. Rapant, K. Slaninova, and J. Martinovic. Traffic incident classification using bayesian approach and markov chains. In *Proceedings of the 13th International Symposium on Operational Research*, pages 143–148. Slovenian Society Informatika, 2015. (SCOPUS, major contribution)

- L. Rapant, T. Martinovic, K. Slaninova, and J. Martinovic. Traffic speed prediction using probabilistic graphical models. In *Proceedings of the 3rd International Conference on Traffic and Transport Engineering (ICTTE)*, pages 941–948. Scientific Research Center LTD Belgrade, 2016. (WoS, major contribution)

- L. Rapant, K. Slaninova, J. Martinovic, and T. Martinovic. Traffic speed prediction using hidden markov models for Czech Republic highways,. In *Proceedings of the 10th KES International Conference on Agent and Multi-Agent Systems: Technology and Applications*, volume 58 of *Smart Innovation, Systems and Technologies*, pages 187–196. Springer Science and Business Media Deutschland GmbH, 2016. (SCOPUS SJR 0.157, WoS, major contribution)

- L. Rapant. Markov Chain Model Approach for Traffic Incident Length Prediction. In *Proceedings of International Conference on Intelligent Traffic and Transportation (ICITT 2017).* (Accepted)(SCOPUS, major contribution)

A.1.2 Proceedings articles

- L. Rapant. Traffic Flow Modelling Based on Sparse Data. In *proceedings of WOFEX 2014*, pages 526–531. VŠB-TUO, 2014, ISBN: 978-80-248-3458-0.

A.2 Unrelated to the Thesis

A.2.1 Journal articles (indexed):

- K. Tomala, J. Plucar, P. Dubec, L. Rapant, M. Voznak. The data extraction using distributed crawler inside the multi-agent system. *Advances in Electrical and Electronic Engineering*, 11(6):455-460, VSB-TUO, 2013. (SCOPUS 0.247, minor contribution)

- P. Dubec, J. Plucar, L. Rapant. Case Study of Evolutionary Process Visualization Using Complex Networks. *Advances in Intelligent Systems and Computing*,210:125-135. Springer Verlag, 2013. (SCOPUS, major contribution)

A.2.2 Journal articles (reviewed):

- M. Kacmarik, L. Rapant. New GNSS tomography of the atmosphere method - proposal and testing. *Geoinformatics*, volume 9, pages 63-76. CVUT, 2012. (major contribution)

- O. Res, J. Stránský, L. Rapant. Lokalizace kostních dlah při ošetření zlomeniny lícně-čelistního komplexu. *Česká Stomatologie*, roč. 117, č.2., pages 43-47. CLS JEP, 2017. (minor contribution)

A.2.3 Proceedings articles (indexed)

- M. Fajkus, J. Nedoma, S. Kepak, L. Rapant, R. Martinek, L. Bednarek, M. Novak, V. Vasinek. Mathematical model of optimized design of multi-point sensoric measurement with Bragg gratings using wavelength divison multiplex. In *Proceedings of Optical Modelling and Design IV*, volume 9889 of *Proceedings of SPIE - The International Society for Optical Engineering*, art. no. 98892F. SPIE, 2016. (SCOPUS 0.228, WoS, minor contribution)

- M. Golasowski, R. Tomis, J. Martinovic, K. Slaninova, L. Rapant. Performance evaluation of probabilistic time-dependent travel time computation. In *Proceedings of 14th IFIP TC 8 International Conference on Computer Information Systems and Industrial Management*, volume 9842 of *Lecture Notes in Computer Science*, pages 377-388. Springer Verlag, 2016. (SCOPUS SJR 0.315, WoS, minor contribution)

- R. Tomis, L. Rapant, J. Martinovic, K. Slaninova, I. Vondrak. Probabilistic time-dependent travel time computation using Monte Carlo simulation. In *Proceedings of the 2nd International Conference on High Performance Computing in Science and Engineering*, volume 9611 of *Lecture Notes in Computer Science*, pages 161-170. Springer Verlag, 2015. (SCOPUS SJR 0.315, WoS, major contribution)

- R. Tomis, J. Martinovic, K. Slaninova, L. Rapant, I. Vondrak.: Time-dependent route planning for the highways in the Czech Republic. In *Proceedings of 14th IFIP TC 8*

International Conference on Computer Information Systems and Industrial Management, volume 9339 of *Lecture Notes in Computer Science*, pages 145–153. Springer Verlag, 2015. (SCOPUS SJR 0.315, WoS, major contribution)

A.2.4 Proceedings articles

- L. Rapant. Ant Colony Optimization for Traffic Routing on a Dynamic Network with Anti-Congestion Measures. *proceedings of WOFEX 2012*, pages 417–422. VSB-TUO, 2012, ISBN: 78-80-248-2769-8.
- L. Rapant. Comparison of Bio-inspired and Mathematical Optimization Methods on a Graph Drawing Problem. In *proceedings of WOFEX 2013*, pages 529–534. VSB-TUO, 2013, ISBN: 978-80-248-3073-5.

B International and national project participation

- RODOS (http://www.centrum-rodos.cz/, 2012-2016, Centrum kompetence TAČR, researcher)
- ANTAREX (http://www.antarex-project.eu/, 2016-present, H2020, researcher)
- NPU II (LQ) (http://www.msmt.cz/vyzkum-a-vyvoj/o-programu-npu-ii-lq, 2016-present, MŠMT ČR, researcher)
- INDECT (http://www.indect-project.eu/, 2012-2014, 7th Framework Programme EU, researcher)
- Matematika pro inženýry 21. století (http://mi21.vsb.cz/, 2012-2013, MŠMT ČR, technical support)
- SP2011/100 Modelování a kvantifikace rizik v lékařství, zpracování biomedicínských dat (2011, VŠB-TUO, researcher)
- SP2012/108 Modelování a kvantifikace rizik (2012, VŠB-TUO, researcher)
- SP2013/116 Analýza rizik pro aplikace v průmyslu a biomedicíně (2013, VŠB-TUO, researcher)
- SP2014/42 Modelování rizik v průmyslu a biomedicíně (2014, VŠB-TUO, researcher)
- SP2015/114 Využití HPC pro analýzu časových řad zatížených neurčitostí (2015, VŠB-TUO, researcher)
- SP2016/166 Využití HPC pro analýzu časových řad zatížených neurčitostí II (2016, VŠB-TUO, researcher)