

Identifying Hidden Influences of Traffic Incidents' Effect in Smart Cities

Attila M. Nagy

Department of Networked Systems and Services
Budapest University of Technology and Economics
Magyar Tudósok krt 2., Budapest, Hungary
Email: anagy@hit.bme.hu

Vilmos Simon

Department of Networked Systems and Services
Budapest University of Technology and Economics
Magyar Tudósok krt 2., Budapest, Hungary
Email: svilmos@hit.bme.hu

Abstract—The road network of big cities is a complex and hardly analyzable system in which the accurate quantification of interactions between nonadjacent road segments is a serious challenge. In this paper we would like to present a novel method able to determine the effects (the time delay and the level of the correlation) of distinct road segments on each other of a smart city's road network. To reveal these relationships, we are investigating unexpected events such as traffic jams or accidents. This novel analysis can give a significant insight to improve the operation of currently widespread traffic prediction algorithms.

I. INTRODUCTION

NOWADAYS smart city services are becoming more widespread than ever as cities are growing and becoming more and more crowded as a result of urbanization and growth of the world population. The rapid progress of urbanization improved life of many people, but also brought remarkable challenges, like traffic congestions that can lead to increased energy/fuel consumption and enormous emission of pollutants [1]. These phenomena heavily and directly impact the health, the life quality and expectancy of city dwellers. According to [2], laboratory studies indicated that transport-related air pollution may increase the risk of developing allergies and can exacerbate symptoms, particularly in susceptible subgroups, while [3] showed that traffic jams increase the risk of heart attacks.

Intelligent management systems, such as Advanced Traffic Management System (ATMS) and Intelligent Transportation System (ITS) can help overcome or significantly reduce the impact of such negative effects on city dwellers. Forecasts of these systems can support traffic control centers in managing the road network and allocating resources systematically, for example opening/closing lanes, pricing dynamically parking places or adapting the traffic lights to the current traffic trends.

In vehicle navigation the knowledge of traffic forecasts for different routes during the route planning is advantageous, as the devices will be able to calculate more efficient routes and reduce travel time. Insight into vehicular flows of smart cities could make searching for parking spaces much easier and faster. It could also provide added value for emerging V2X based traffic control systems, which can play an important role in route planning of self-driving cars.

In the literature, there are numerous prediction models utilized for traffic flow prediction, however the road network of

big cities is a complex and hardly analyzable system in which unexpected events could significantly decrease the correctness of the result of the prediction models.

Every predicted value is composed of a predictable component and an error [4], which includes both prediction error and unpredictability of uncertainty. Thus the predictable value is derived from the deterministic part and the predictable part of uncertainty. Therefore it follows, that the predictability of the traffic flow depends on whether the model is able to predict the uncertainty part or not. Fortunately, lots of prediction models can be prepared for handling the uncertainty part by integrating different external data sources. The uncertainty is influenced by many factors, like weather condition, mass events, road constructions, road closures, accidents etc. By considering these external environmental factors, the error of prediction models can be decreased.

In this paper, a new method will be presented, which aims to reduce the previously mentioned uncertainties. The algorithm focuses on unexpected events, such as traffic accidents, which can have a negative impact on the traffic prediction. By investigating the effect of these phenomena, the algorithm is able to:

- identify neighboring road segments that could be affected by the event
- to determine the level of correlation between the road segment affected by the accident and its neighboring road segments,
- to quantify the time delay: the time needed for the effects of the accident to propagate to neighboring road segments.

By fusing this information with real-time traffic information, the prediction models will be able to provide more accurate predictions even in the case of an unexpected event happening nearby.

The paper is organized as follows. In Section II., various prediction techniques are introduced aimed at reducing the error of prediction models. This section also contains a short summary of usable data sources. In Section III, the algorithm is presented in detail, which is followed by a case study simulation in Section IV. Section V concludes the paper and points out the current weaknesses and future improvements of the algorithm.

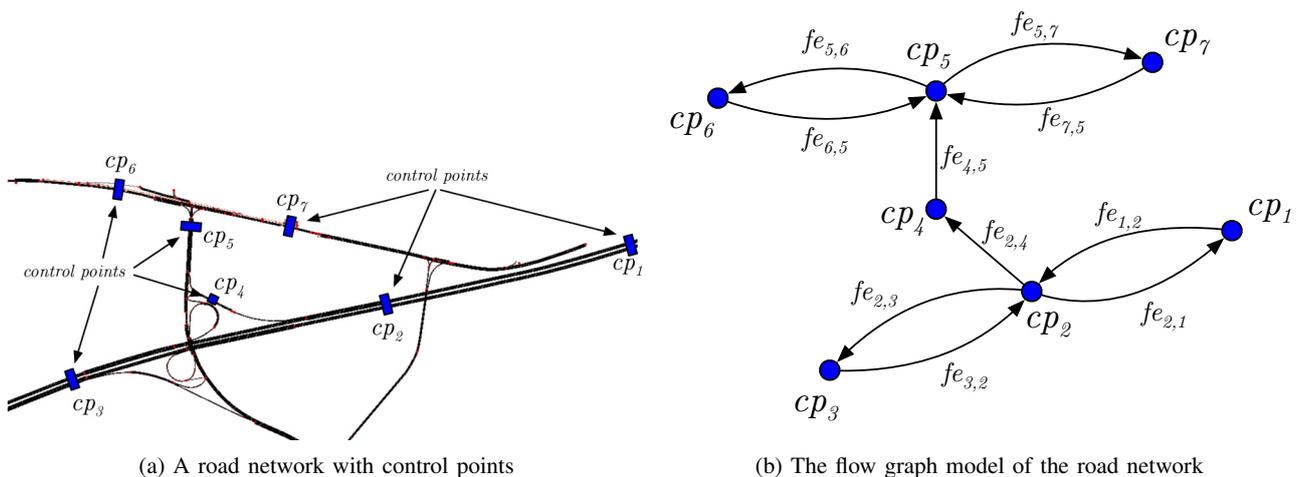


Fig. 1: An example of the traffic graph model interpretation. On Figure 1a, the original road network is depicted with control points, while Figure 1b shows the road network's graph representation.

II. TRAFFIC PREDICTION METHODS IN SMART CITIES

A. Prediction Techniques

There are several proposed prediction models [5], which perform well for regular conditions, however an unexpected event could significantly decrease the quality of their traffic forecasts. The reason is that the predictability of the traffic flow depends on whether the model is able to predict the uncertainty part of the traffic flow with the required precision. The uncertainty is influenced by many factors, like weather, events, road constructions, lighting conditions etc. Incorporating external environmental factors and fused data [6] to the model is crucial to decrease the error of the prediction and increase the predictable part of uncertainty.

Traffic is predictable in the sense that it does not vary significantly during weekdays and during most months of the year [7]. In [8] similar results were obtained as well as they have found a relatively high daily predictability of traffic conditions despite the absence of any apriori knowledge of drivers' origins and destinations and the quite different travel patterns between weekdays and weekends.

A neurowavelet prediction algorithm was proposed [9] to forecast the hourly traffic flow considering the effect of rainfall. In the article, the authors use a stationary wavelet transform to reveal correlation between different weather conditions and changes in the traffic flow. An examination was carried out [10] whether or not road usage on a particular location determines the impacts of various weather conditions. The study showed that the precipitation, cloudiness and wind speed reduce traffic intensity, while high temperatures and hail significantly increase traffic intensity.

Other papers concentrated on the spatio-temporal property of the traffic flow, by using different Autoregressive Integrated Moving Average (ARIMA) model variants [11]–[13], applying K-Nearest Neighbors (KNN) models [14], [15] while others employed Convolutional Neural Network (CNN) for this pur-

pose [16]–[18]. Deep learning based prediction model was also presented for spatio-temporal data [17]. The prediction model uses spatial and temporal relations and integrates global information (such as day of week, meteorological conditions, etc.) to decrease the uncertainty.

Relation between traffic predictability and prediction time horizon was investigated [4] by examining spatio-temporal traffic relationship using Cross-correlation function (CFF). The time lag calculated by CFF can be used to determine prediction time horizon, and the cross-correlation coefficient can be utilized to identify the spatial relations that can be used in prediction.

Solutions enumerated in this section aim to handle the previously listed uncertainties of traffic flows. However, we have not found significant work aimed at increasing the predictability of traffic flow through the investigation of unexpected events like accidents or traffic congestions of uncontrolled traffic flow. In this paper, we will present a novel method, which can exploit these events to measure the time delay and calculate the level of influence between distinct road segments.

B. Data sources

In the first generation of ATMSs and ITSs [19] the utilized data sources were various presence sensors in fixed positions, able to detect the presence of nearby vehicles. Initially inductive loop detectors were the most popular, but nowadays a wide variety of sensors became available [20].

Recently, the advent of GPS equipped smartphones and vehicles has given rise to a relatively new type of data source that could supplement presence type sensors to gather more detailed information or get data about roads, which have not been covered with presence sensors yet.

Our method can leverage both types of data sources, but in the case of GPS traces, a preprocess step is needed to be inserted before the data model building.

III. METHODOLOGY

The road network of big cities is a complex and hard to analyze system in which the accurate quantification of interactions between nonadjacent road segments is almost an insolvable objective, because of the unique decisions of thousands of drivers which makes the interactions invisible. However an unexpected event could be used to reveal hidden connections between road segments, because they always appear as an outlier in the timeseries of the investigated data type (traffic speed, traffic flow count, travel time, etc.). It follows, that if the emergence of an unexpected event is known, the ripple of that event can be observed through the network, thus the hidden correlations will become observable. As an analogy, we suppose that one can think of a road network as a huge black box system, which has one input and numerous outputs. If the system is fed with an unusual input, the inner behavior of the system can be inferred through the outputs.

In this section, the Algorithm for Identifying Hidden Influences (AIHI) will be introduced in detail, which targets to exploit unexpected traffic events to reveal the previously mentioned connections. In Subsection III-A, the graph based data model will be presented which is suitable for the analysis of unexpected events, then the distinct steps of the algorithm will be explained in Subsection III-B.

A. Data model

In our solution, the road network is modeled with a $FG = (CP, FE)$ directed flow graph. Each $cp \in CP$ node represents a control point, which is a special point of the road network, where the traffic measurement is feasible by different type of traffic sensors (such as inductive loop detectors, radar sensors, audio sensors, etc.). A cp control point itself does not store any traffic data, they just measure the traffic flow at their position.

The $fe \in FE$ directed flow edges represent a link between two adjacent control points (cp_{src}, cp_{dst}), where there is at least one lane between the two control points in the spreading direction of the traffic. To every fe directed flow edge, a ts time series is assigned, which stores historical traffic flow data. The ts time series of fe flow edge will contain those measurements, which are provided by the cp_{dst} destination control point of fe . For instance, if there is a directed flow edge between cp_1 (source) and cp_2 (destination) control points denoted by $fe_{1,2}$, then the $fe_{1,2}$ edge will contain the measurements of cp_2 control point.

Besides the data from fixed position sensors, the data model is also able to utilize GPS traces, if virtual control points are defined and trace data is aggregated in these points.

On Figure 1, an example of the traffic graph interpretation is depicted, it shows how the data model have to be interpreted on a simple road structure. Figure 1a illustrates a simple road network with control points, which are marked by cp_i identifiers. On Figure 1b, the directed flow graph of the previous road network is visualized. The cp_i identifiers on this figure are identical with the identifiers of the ones on Figure 1a, and there are $fe_{x,y}$ directed flow edges in the graph, if

cp_x and cp_y control points are connected in the road network in the spreading direction of the traffic.

We also have to deal with the timeliness of our model. The different fixed position or GPS sensors sample the measured data type with different frequencies based on their settings. Consequently the traffic analysis requires a homogeneous sampling frequency. Different aggregating time intervals are used in the literature for this purpose, which mainly depend on the task at hand.

Generally, narrow intervals, for example 10 seconds, are meaningless and really noisy. We have found that the most common time intervals are in minutes (5-10 minutes) [21], [22], but there are also many papers claiming that longer time intervals would be more effective, like quarter or half an hour [16], [23]. For our model, we chose a 30 seconds time interval because too long time intervals could hide important features of an unexpected event and the noisiness of the 30 seconds scale does not affect the correctness of the AIHI.

B. The steps of the algorithm

1) *Initialization:* Let us denote a time series of $fe_{src,dst}$ flow edge by $ts_{src,dst}$ in which data can be described as an ordered sequence of discrete measurements:

$$ts_{src,dst} = \{m_t\} \quad t = 1, 2, \dots, T \quad (1)$$

where m_t is the measured traffic count value at time t at the cp_{dst} control point.

The entry point of the algorithm will be an fe directed flow edge, for which the associated ts time series shows unexpected event between an arbitrary (t_{start}, t_{end}) time interval. The (t_{start}, t_{end}) time interval contains an unexpected event, if a statistically significant change can be detected in the behavior/shape of the ts time series between t_{start} and t_{end} compared to the ts_{hist} historical average, or in other words, ts time series contains an anomalous part compared to the ts_{hist} historical average.

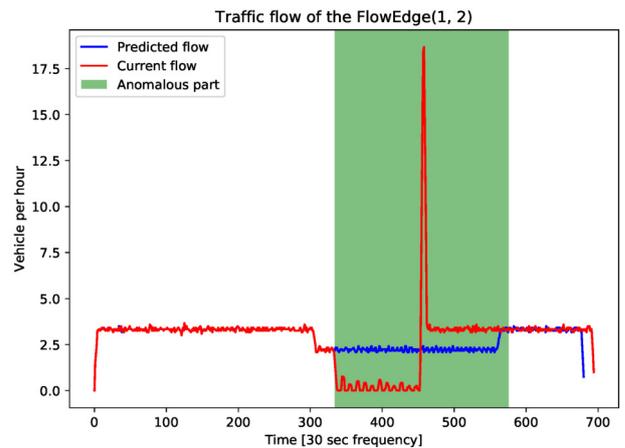


Fig. 2: Visualization of anomaly behavior compared to Predicted flow

Input :

- FG : Directed flow graph of the road network
- fe : The investigated directed edge
- t_{start} : The start time of the unexpected event

Output: The effects of the event organized in a tree structure

```

1  job_pool ← [];
2  visited_edges ← [];
3  ts_an = findAnomaly(fe.ts, t_start);
4  addJob({fe, ts_an, t_start, forward});
5  while job_pool is not empty do
6    j = getNext(job_pool);
7    addVisitedEdge(j.fe);
8    (b, i) = bestFitLag(j.ts_an, j.fe.ts, j.t_start,
9    j.dir);
10   is_anomaly, next_ts_an =
11   findAnomaly(j.ts, nj.t_start + b);
12   if is_anomaly then
13     s_neighs = getEdges(j.fe.cp_src);
14     foreach edge in s_neighs do
15       if j.fe.cp_src is edge.cp_dst and edge not
16       in visited_edges then
17         addJob({edge, next_ts_an,
18         t_start + b, backward});
19       end
20     end
21     d_neighs = getEdges(j.fe.cp_dst);
22     foreach edge in d_neighs do
23       if j.fe.cp_dst is edge.cp_src and edge not
24       in visited_edges then
25         addJob({edge, next_ts_an,
26         t_start + b, forward});
27       end
28     end
29   end
30 end

```

Algorithm 1: AIHI algorithm

There are two methods to discover such time intervals. The easiest way when we have apriori knowledge about the occurred unexpected events like accidents or road closures as these can be used directly as the input of the AIHI algorithm. The other possible approach is to execute an extensive search in the raw historical dataset for unexpected events. This approach requires a classification model that is able to decide whether a $ts_{src,dst}$ time series contains an unexpected event or not.

The AIHI algorithm needs the following three inputs:

- The FG flow graph of the road network
- The fe flow edge, which was the source of the unexpected event
- The t_{start} time of the emergence of the unexpected event on fe

Utilizing these three inputs, the algorithm will follow the

effect of the unexpected event through the traffic flow graph and determining those flow edges that could be affected by the input event.

2) *Processing of a job*: AIHI algorithm 1 uses a job pool based approach in which the whole investigation task is separated to smaller independent subtasks. In this case, a subtask is responsible for the investigation of whether the currently examined fe flow edge's ts time series is affected by the source unexpected event or not. To run a job, the following elements are necessary:

- An fe flow edge
- The ts_{an} time series, which contains the whole anomaly part identified by the source job
- The t_{start} emergence time of the anomaly in the source job
- The dir direction of the spread of the event (forward or backward), because the unexpected events of the road network can have an effect in both directions.

To start the algorithm, the first job will be created from the entry point. The entry point contains all necessary job input parameters, except the ts_{an} time series, thus by using the findAnomaly function (in Section III-B4), the anomalous part of the ts time series of fe flow edge have to be calculated. After that, the execution of the algorithm can be started and the processing of jobs will be continued until the pool has been emptied.

A job will execute these steps:

- 1) Find the best fitting $best_lag$ time lag between the ts time series of fe flow edge and ts_{an} anomalous time series from t_{start} in the chosen dir direction (see Algorithm 2)
- 2) Check that an anomalous event can be identified from $best_lag$ or not (see Algorithm 3)
- 3) If an anomalous event is detected during the second step, then the adjacent flow edges of the investigated fe flow edge are put into the job pool as new jobs in which the source job will be the current job

3) *Find best lag*: The BestFitLag function is responsible for determining the start of an anomalous event in the ts time series of the current $job.fe$ flow edge.

We can assume that the shape of the anomalous time series ts_{an} is quite similar to the anomaly observed in the actual ts time series of $job.fe$. To measure this similarity we defined a new distance function, called shape_dist (Equation 2). Contrary to other distance function like Manhattan, Euclidean or DynamicTimeWarping [24], it measures the similarity the time series' shape by differentiating changes in the different time series:

$$shape_dist(x, y) = \sqrt{\sum_{i=1}^n ((x_i - x_{i-1}) - (y_i - y_{i-1}))^2} \quad (2)$$

The shape_dist function is calculated with increasing $lag = 0, 1, 2, \dots$ between ts time series and ts_{an} time series. It can

Input :

- ts_{an} : The time series containing the whole anomaly part
- ts : The time series of the fe flow edge
- t_{start} : The emergence time of the anomaly in the time series
- dir : The direction of the spread of the event (forward or backward)

Output:

- $bestlag$: The best fitting time lag
- $influence$: The level of influence

```

1 last ← dist(tsan, shift(ts, tstart));
2 if dir is forward then
3   | i ← 1;
4 end
5 if dir is backward then
6   | i ← -1;
7 end
8 lastlag ← 0;
9 while dist(tsanom, shift(ts, tstart + i))
  ≤ last do
10  | lastlag ← lastlag + i;
11 end
12 bestlag = i;
13 length = len(tsan);
14 influence =  $\frac{1}{1 + \exp(\frac{1}{length} * last)}$ ;

```

Algorithm 2: BestFitLag

be assumed, that while the delay is increasing, the distance between the two time series will decrease until the best fit is reached. Thus if the calculated distance values start to increase, the possible best delay has been reached, because the adjacent road segments show high correlation in general. However sometimes the calculated delay can be just a local minimum, thus a simulate annealing is applied to find the global optimum.

Furthermore, the best distance value can be used to express the influence between the two flow edges, however a transformation is required, converting the $shape_dist$ function's $[0, \infty)$ domain to $[1, 0]$ domain. Higher values mean a stronger influence, while lower values mean a weaker influence. Equation 4 is designed for this purpose:

$$distance = shape_dist(ts_{an}, shift(ts, lastlag)) \quad (3)$$

$$influence = \frac{1}{1 + \exp(\frac{1}{length} * distance)} \quad (4)$$

, where the $length$ parameter equals with the length of ts_{an} .

4) *Find anomaly*: The $findAnomaly$ function (Algorithm 3) is responsible for determining whether an anomalous part starting from t_{start} can be identified. If an anomalous part is found, the function also returns its length.

The $findAnomaly$ function exploits the observation, that the error of a prediction model significantly increases when

Input :

- ts : The time series of the fe flow edge
- ts_{hist} : The historical average time series of the fe flow edge
- t_{start} : The emergence time of the anomaly in the time series

Output:

- $is_anomaly$: The input ts time series contains anomaly from t_{start} or not
- $next_ts_{an}$: The anomaly part in the input ts time series if it exists

```

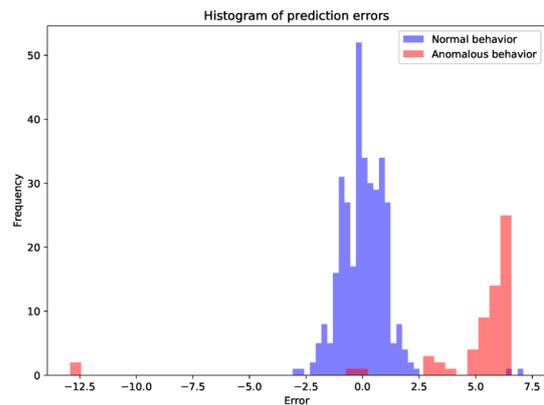
1 model ← exp_smooth(tshist);
2 error ← measure_error(model, tshist);
3 error_dist ← norm_dist(error);
4 length ← 0;
5 while ts[tstart + length] from error_dist do
6   | length ← length + 1;
7 end
8 if length is 0 then
9   | is_anomaly ← false;
10 else
11   | is_anomaly ← true;
12 next_tsan ← ts[tstart, tstart + length]

```

Algorithm 3: findAnomaly

anomalous behavior can be observed. The error of prediction was analyzed and we could conclude, that the prediction error has a normal distribution in case of normal behavior (depicted on Figure 3), while the distribution of the error significantly differs in case of anomalous behavior. Thus the operating principle of $findAnomaly$ is:

- 1) Fitting a prediction model on ts_{hist}
- 2) Running until the prediction error of the model returns to a normal distribution


Fig. 3: Histogram of different prediction errors

Therefore for $findAnomaly$, a prediction model is constructed by applying a simple moving average on ts_{hist} , then

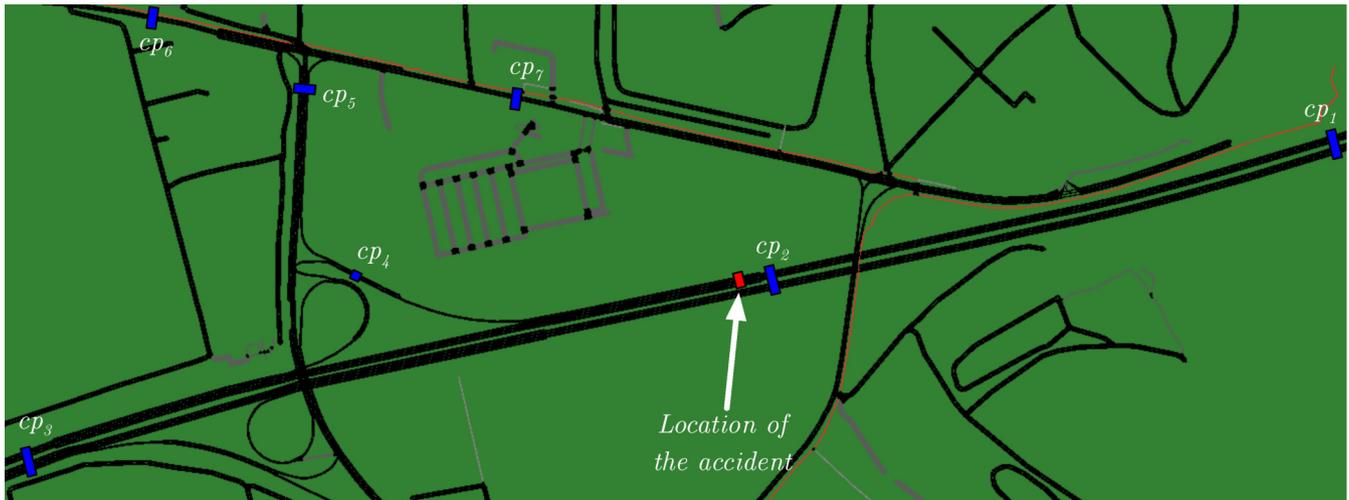


Fig. 4: The simulation map with location of traffic accident and control points

based on that model the distribution of the prediction error is determined. The algorithm starts from t_{start} and runs until the prediction error of the model returns to a normal distribution. This point in the time series is $length$ apart from t_{start} , therefore the length of the anomaly can be calculated. If the $length$ equals with zero, it means that ts does not contain an anomaly.

IV. CASE STUDY

After the implementation of the AIHI, in this section we will demonstrate its ability to follow the spread of the effect of a traffic accident through a traffic flow graph.

At first, we searched for datasets containing traffic flow data and traffic accidents as well, but unfortunately there was no such publicly available traffic flow dataset at the time of writing. Because of this, a simulation framework was used for the evaluation.

As the simulation framework, we chose Simulation of Urban Mobility (SUMO), a free and open traffic simulation suite, which is available since 2001. SUMO allows modeling of intermodal traffic systems, including roads, vehicles, public transport and pedestrians. Included with SUMO is a wealth of supporting tools, which handle tasks such as route finding, visualization, road network import from open street maps and emission calculation.

In our scenario, besides the real traffic flow data, a traffic accident had to be generated. The authors of [25] used traffic lights for this purpose, thus this approach had been applied in our simulation as well. The accident is simulated by opening only one of the four available lanes on a road.

In the simulation, high rank roads of Budapest's suburb are examined. Seven control points (using inductive loop detectors) were placed on the map as depicted on Figure 4. We simulated five hours of traffic flow, which was similar to afternoon rush hours. The vehicles are simulated with $speedFactor = normc(1, 0.1, 0.2, 2)$, which means that 95%

of the vehicles drove between 80% and 120% of the legal speed limit. In the beginning we simulated normal traffic flow, then an one hour long traffic accident had been inserted after two and a half hours near to cp_2 , so the effect of the accident could be identified first on $fe_{1,2}$ flow edge (Figure 2).

As mentioned in Subsection III-B1, the entry point of AIHI was $fe_{1,2}$ flow edge with t_{start} start time of the simulated traffic accident. The result of AIHI, is displayed on Figure 5. The spreading tree of the accident shows that the farther control points were, the bigger the detected time lags became, while the influences were decreasing as expected. The change of the pattern of the anomalous part was also visualised between cp_4 and cp_5 on Figure 6.

V. CONCLUSION

The road network of big cities is a complex and hard to analyze system in which the accurate quantification of interactions between nonadjacent road segments is almost an insolvable objective, because of the unique decisions of thousands of drivers which makes the interactions invisible. In this paper a novel algorithm (AIHI) has been presented, that is able to exploit unexpected traffic events to reveal the hidden connections between nonadjacent road segments and provides the following information:

- identify nearby road segments that could be affected by the event
- if a road segment is affected, the exact level of influence between the affected and accident road segments,
- the time delay: the time between the event and its detection on the affected road segments

Combining these new information sources with real-time traffic information, the accuracy of prediction models able to integrate external environmental variables can be increased.

We have demonstrated the capabilities of AIHI with simulations on a real road network which results can be depicted as spreading tree of the accident.

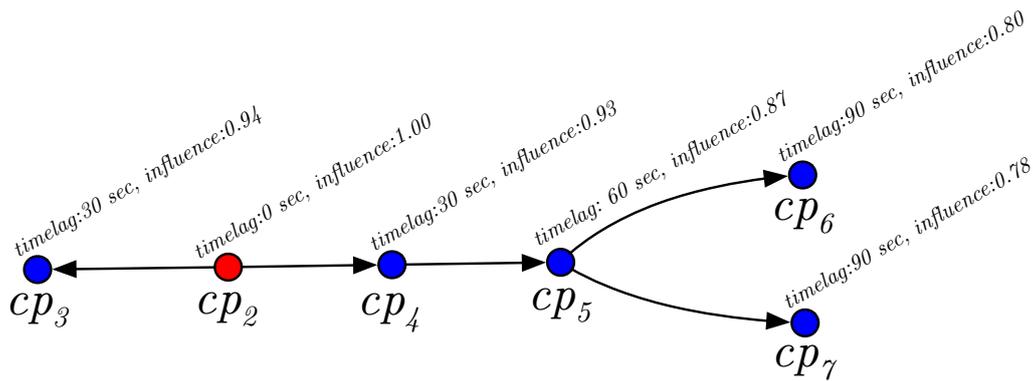
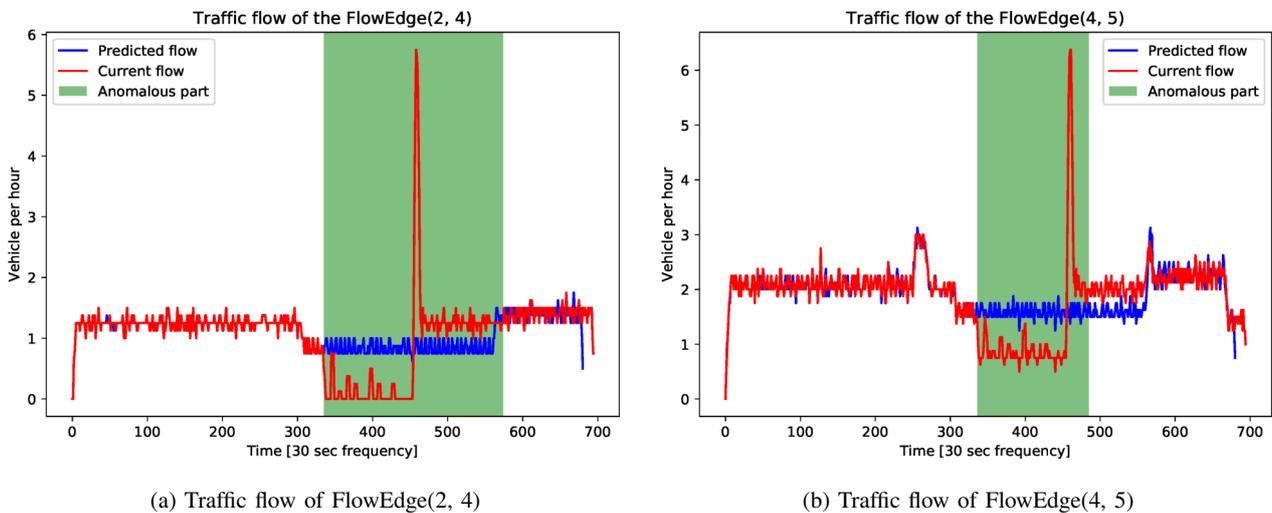


Fig. 5: Visualization of the AIHI's result. The examined control point was cp_2 (red dot), where the time lag is zero and the influence is 1.0. Other influenced control points in the road network identified by AIHI are marked with blue dots.



(a) Traffic flow of FlowEdge(2, 4)

(b) Traffic flow of FlowEdge(4, 5)

Fig. 6: On Figure 6a and Figure 6b the measured traffic flows are depicted at cp_4 and cp_5 , respectively. On Figure 6b the effect of other connected roads can be seen, where the anomalous part starts to differ compared to the anomalous part of Figure 6a

In the future, we are planning to extend prediction models like Neural Networks (NN) and KNN to utilize the result of AIHI. We also want to develop a classification model able to decide whether a time series contains an unexpected event or not, because the input of parameters of AIHI are currently determined manually. Applying the classification model, the determination of AIHI's input parameters will be automated.

VI. ACKNOWLEDGMENT

The research reported in this paper was supported by the BME- Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC).

REFERENCES

[1] J. He, S. Gong, Y. Yu, L. Yu, L. Wu, H. Mao, C. Song, S. Zhao, H. Liu, X. Li, et al., Air pollution characteristics and their relation to meteorological conditions during 2014–2015 in major chinese cities, *Environmental pollution* 223 (2017) 484–496.

[2] M. Krzyżanowski, B. Kuna-Dibbert, J. Schneider, Health effects of transport-related air pollution, WHO Regional Office Europe, 2005.

[3] A. Peters, S. Von Klot, M. Heier, I. Trentinaglia, A. Hörmann, H. E. Wichmann, H. Löwel, Exposure to traffic and the onset of myocardial infarction, *New England Journal of Medicine* 351 (17) (2004) 1721–1730.

[4] Y. Yue, A. G. Yeh, Y. Zhuang, Prediction time horizon and effectiveness of real-time data on short-term traffic predictability, in: *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, IEEE, 2007*, pp. 962–967.

[5] A. Ermagun, D. Levinson, Spatiotemporal traffic forecasting: review and proposed directions, *Transport Reviews* (2018) 1–29.

[6] N.-E. El Faouzi, H. Leung, A. Kurian, Data fusion in intelligent transportation systems: Progress and challenges—a survey, *Information Fusion* 12 (1) (2011) 4–10.

[7] A. Stathopoulos, M. Karlaftis, Temporal and spatial variations of real-time traffic data in urban areas, *Transportation Research Record: Journal of the Transportation Research Board* (1768) (2001) 135–140.

[8] J. Wang, Y. Mao, J. Li, Z. Xiong, W.-X. Wang, Predictability of road traffic and congestion in urban areas, *PLoS one* 10 (4) (2015) e0121825.

[9] S. Dunne, B. Ghosh, Weather adaptive traffic prediction using neu-

- rowavelet models, *IEEE Transactions on Intelligent Transportation Systems* 14 (1) (2013) 370–379.
- [10] M. Cools, E. Moons, G. Wets, Assessing the impact of weather on traffic intensity, *Weather, Climate, and Society* 2 (1) (2010) 60–68.
- [11] Q. T. Tran, Z. Ma, H. Li, L. Hao, Q. K. Trinh, A multiplicative seasonal arima/garch model in evn traffic prediction, *International Journal of Communications, Network and System Sciences* 8 (04) (2015) 43.
- [12] W. Min, L. Wynter, Real-time road traffic prediction with spatio-temporal correlations, *Transportation Research Part C: Emerging Technologies* 19 (4) (2011) 606–616.
- [13] S. V. Kumar, L. Vanajakshi, Short-term traffic flow prediction using seasonal arima model with limited input data, *European Transport Research Review* 7 (3) (2015) 21.
- [14] P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding, J. Sun, A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting, *Transportation Research Part C: Emerging Technologies* 62 (2016) 21–34.
- [15] B. Yu, X. Song, F. Guan, Z. Yang, B. Yao, k-nearest neighbor model for multiple-time-step prediction of short-term traffic condition, *Journal of Transportation Engineering* 142 (6) (2016) 04016018.
- [16] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang, Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction, *Sensors* 17 (4) (2017) 818.
- [17] J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi, Dnn-based prediction model for spatio-temporal data, in: *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ACM, 2016, p. 92.
- [18] Y. Li, R. Yu, C. Shahabi, Y. Liu, Graph convolutional recurrent neural network: Data-driven traffic forecasting, arXiv preprint arXiv:1707.01926.
- [19] B. Singh, A. Gupta, Recent trends in intelligent transportation systems: a review, *Journal of Transport Literature* 9 (2) (2015) 30–34.
- [20] Y. Yue, Traffic sensors, *International Encyclopedia of Geography: People, the Earth, Environment and Technology: People, the Earth, Environment and Technology* (2016) 1–7.
- [21] R. Fu, Z. Zhang, L. Li, Using lstm and gru neural network methods for traffic flow prediction, in: *Chinese Association of Automation (YAC), Youth Academic Annual Conference of, IEEE*, 2016, pp. 324–328.
- [22] D. Xia, B. Wang, H. Li, Y. Li, Z. Zhang, A distributed spatial-temporal weighted model on mapreduce for short-term traffic flow forecasting, *Neurocomputing* 179 (2016) 246–263.
- [23] Y. Tian, L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network, in: *Smart City/SocialCom/Sustain-Com (SmartCity)*, 2015 IEEE International Conference on, IEEE, 2015, pp. 153–158.
- [24] M. Müller, Dynamic time warping, *Information retrieval for music and motion* (2007) 69–84.
- [25] J. Q. Shi, L. Cheng, Simulation and analysis of highway traffic accident based on vissim, in: *Applied Mechanics and Materials*, Vol. 253, Trans Tech Publ, 2013, pp. 1682–1685.