Real-time Public Transport Delay Prediction for Situation-aware Routing

Lukas Heppe[1] and Thomas Liebig[1]

TU Dortmund University
Computer Science VIII: Artificial Intelligence Unit
{lukas.heppe, thomas.liebig}@tu-dortmund.de
http://www-ai.cs.uni-dortmund.de/

Abstract. Situation-aware route planning gathers increasing interest. The proliferation of various sensor technologies in smart cities allows the incorporation of real-time data and its predictions in the trip planning process. We present a system for individual multi-modal trip planning that incorporates predictions of future public transport delays in routing. Future delay times are computed by a Spatio-Temporal-Random-Field based on a stream of current vehicle positions. The conditioning of spatial regression on intermediate predictions of a discrete probabilistic graphical model allows to incorporate historical data, streamed online data and a rich dependency structure at the same time. We demonstrate the system with a real-world use-case at Warsaw city, Poland.

1 Introduction

With the emergence of smart cities, trip computation received increased attention. While conventional trip computation algorithms minimize a static cost function and provide an optimal route for an unlikely stationary traffic situation with constant costs. Traffic situations are not stationary but vary over time, e.g. at rush hour commuters cause traffic jams at streets which are almost empty at night. The integration of various sensor systems (e.g. crowdsourcing, video cameras, automatic traffic loops, [20]) in the smart city ecosystem enables incorporation of real-time measurements in intelligent traffic systems, and their predictions [21].

In this work, we target, for the first time, the question how to incorporate predictions of delays in the public transport network in multi-modal trip planning. In result, we aim to obtain a smart trip planner that supports citizens of a smart city to make informed decisions on their transit route. The possible benefits for the informed travelers are:

1. A smart decision among different modes of transportation,
2. a smart choice among different transit routes,
3. an informed decision among different initial walking directions, or
4. different transit stops.

We exemplify points two and three next in Warsaw, the capital of Poland, for different representative cases, see Figure 1. In the two subfigures on the top

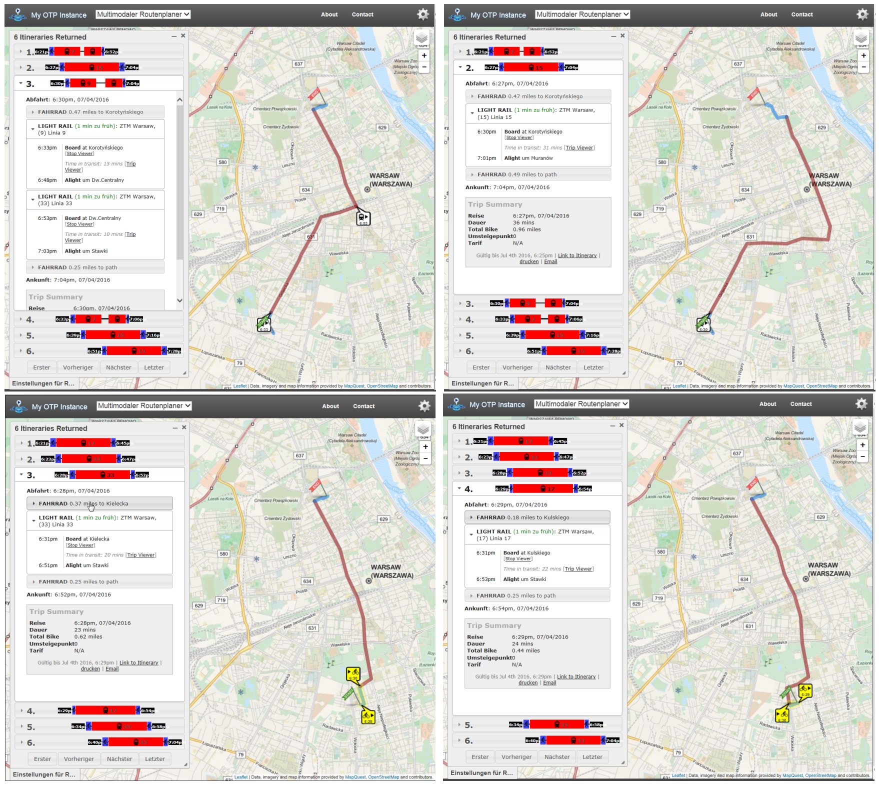


Fig. 1: Exemplified trips for same start and goal. Top: different tram lines are suggested, bottom: different initial walking direction is suggested. Best viewed in color.

same origin destination pairs lead to different transit suggestions. Different initial walking directions are suggested in the lower subfigures of Figure 1.

As seen in previous examples, the prediction of delay supports planning of situation-aware trips in advance (not just when travelling). This enables a smart decision for the very first step and even enables decision to start a trip earlier or later (depending on expected transfer reachability). Imagine for example, dining with your friends in the suburbs of your city. On your return, our trip planner provides you with the information that the required transfer in the city centre (from the tram in the suburbs to your means of transportation) is likely not to be reached (Note that we incorporate predictions based on the current situation in contrast to existing planners that incorporate just current information). Based on this information, you may stay longer with your friends instead of useless waiting outside at the tram station.

Our approach towards this situation-aware trip planner detects current and past delays of transit vehicles based on a comparison of their live GPS streams with the scheduled arrival times. Of course, other sensor technology e.g. Bluetooth would have been also possible [23,13], but with stationary sensors you easily get problems of sensor placement [16] and a stream of GPS data from the vehicles is available in the city of Warsaw. The detected delays are used to estimate future delays by a probabilistic graphical model. These real-time predictions are incorporated in route computations generated with OpenTripPlanner an open source trip planning tool. The data, our approach bases on, are

- the street network,
- public transport schedules and
- a real-time stream of the current vehicle positions.

We perform our experiments in Warsaw, Poland, and use open data provided via open geospatial consortium standardized protocols and interfaces.

Our paper is structured as follows. Section 2 reviews current state-of-the-art for routing algorithms and positions our work. Afterwards, we present the real-time architecture of our approach that uses predictions-as-a-service. In Section 4 we present the application of an existing Spatio-Temporal-Random-Field (STRF) model to the real-time tram delay prediction task. In section 5 we highlight the application of our approach and discuss future directions for improvement in the closing Section 6.

2 Related Work

The task to plan a route from one start location to a target location is called trip planning, when multiple means of transportation (also called 'travel modes') are involved this becomes multi-modal trip planning. The integration of transportation systems with personal constraints, residential and city services systems can offer real promise for implementing an intelligent transportation infrastructure that can efficiently address issues beyond congestion, resiliency and safety. Trip planning operates on a graph representation of the road and transit network the so-called traffic network G consisting of vertices V (e.g. junctions) and connecting edges E (e.g streets). A cost function maps each edge to a positive number that denotes how much it would 'cost' to travel the corresponding segment. The cost function needs to be homogeneous throughout the traffic network, but can be defined in several ways, such that it holds the most important aspects: for example length of the segment, travel time, or comfortableness. With a given start and end location in the traffic network, trip planning searches the path that connects start and goal and minimizes the cost.

Several algorithms exist to compute this minimizing path. Dijkstra [5] proposes a best-first traversal of the graph where the candidates for traversal are hold in a priority-queue. In the slightly modified version of the algorithm A* [9] the order in the priority-queue for the traversal not only depends on the cumulated costs to reach a vertex in the graph but also on the expected costs to reach the goal from this vertex. Bound by Minkowski's inequality, whereas $||x+y||_p \leq ||x||_p + ||y||_p$ (known as triangle inequality for $p = 2$), A* prunes the search space in comparison to Dijkstra's Algorithm. A sound heuristic for the remaining cost estimation is the geographical distance that is always lower than the road-based distance. In multi-modal trip planning multiple of these traffic networks G (one for each mode) are linked together at locations where it is possible to switch from one mode to another (transfer vertices). Multi-modal trip planning requires a consistent cost function which is applicable to all parts of the traffic network and thus to all modes of transportation.

In case of static cost functions contraction hierarchies [7] are a data structure that speeds-up the A* algorithm and enables trip calculation in large traffic networks at European scale. Instead of searching the shortest path directly within the traffic network, contraction hierarchies reduce the search space to the most important ones. In a preprocessing step these important segments are identified (based on the topology) and the network is extended by edges between these important links.

For transit networks, Transfer Pattern [1] provides a speed-up heuristic. Transfer Pattern exploit that a transit network consists of central locations (hubs as major airports or train stations) where most people from a particular region have to change the means of transportation. These (multi-modal) routing heuristics are great for trip computation in embedded devices, and according to [2] they provide sufficient accuracy in case of dynamic cost functions (based on estimations and predictions of traffic). However, dynamic transfer patterns [14] incorporate also unexpected novel transfers that were enabled by the delay itself.

In this work, we focus on the incorporation of dynamic cost estimates in multi-modal trip planning. Thus, we combine a real-time prediction of delay in transit networks with the trip computation. Previous works introduce already the incorporation of traffic predictions in vehicular path finding. E.g. the work in [15] proposes situation-aware routing with real-time predictions. Their method bases on a spatio-temporal graphical model that provides estimates for future traffic values based on current and past observations. These spatio-temporal estimates serve as cost function for routing and traffic jams were avoided. The work in [19] uses Conditional Random Fields for future traffic prediction, but lacks the inclusion in the trip planning application.

In contrast to vehicular traffic, trams and trains can not overtake, and vehicles in transit networks wait for each others (e.g. connecting trains), this causes delays to propagate differently than vehicular traffic jams. In addition, two modes of transportation may share the same physical resource (e.g. buses or trams riding on vehicular street). Thus, two forms of delays in transit networks are distinguished in literature: 1) a vehicle is late due to own reasons, and 2) other vehicles are late caused by the former [18].

Several models for transit delays are reported in literature. The work in [4] assumes independence. In contrast, [8] allows delays to cumulate. Sophisticated models incorporate dependencies among the vehicles into the delay [11]. In the trip planning application it is a crucial requirement to the prediction model to provide real-time predictions. Thus, we highlight two recent works on delay prediction and delay recognition: [6] applies queueing theory and assumes delays to aggregate, [24] detects delays and unexpected vehicle movement in real-time from the GPS traces.

In contrast, our approach will be a probabilistic one, where similar to the approach in [6] the delay of a vehicle at the stops in a trip depends on its predecessors and the delay event that a vehicle is delayed is detected directly from its GPS stream [24] using spatio-temporal constraints, in the experiments section we compare our approach to [6].

3 Architecture

Our proposed system comprises two layers (1) a real-time event detection layer that processes the incoming GPS data stream of the transit vehicles (detection of delay events and estimation of future delays), and (2) an asynchronous trip planning layer which is triggered by user-generated trip queries and incorporates current predictions[1].

In the event detection layer, every single GPS data is processed and current delays of the vehicles are detected, furthermore this information is used to update (in real-time) predictions of the expected delay for the whole day. The survey in [22] provides a list of possible spatio-temporal event detection and pattern matching frameworks depending on the required expressiveness of the spatio-temporal pattern. We decide to use a streams framework and pose spatio-temporal constraints as real-time operators to the stream of GPS data points.

The asynchronous trip planning layer incorporates the predictions as a service and utilizes them for multi-modal trip computations. In result, we obtain situation-aware routes. Similar to [15], we base our trip planning on the OpenTripPlanner (OTP) implementation. This open source routing software provides interfaces for inclusion of transit schedules (in the commonly used General Transit Feed Specification (GTFS) standard[2]) and OpenStreetMap.

3.1 OpenTripPlanner

OpenTripPlanner (OTP) is an open source initiative for multi-modal route computation. The traffic network for route computation is generated using open data from OpenStreetMap and public transport schedules (in the widely used Google Transit Feed Standard protocol). Thus, OpenTripPlanner is an open source trip planner that connects to open data and provides route calculation capabilities for multiple modes of transportation (e.g. walking, transit) and their combinations.

3.2 Streams Framework

We use TU streams framework as real-time engine [3]. It contains basic real-time machine learning algorithms and provides any-time predictions-as-a-service

[1] Our source code and the required virtual machine are publicly available as vagrant box at `https://bitbucket.org/tliebig/developvm`.

[2] https://developers.google.com/transit/gtfs/

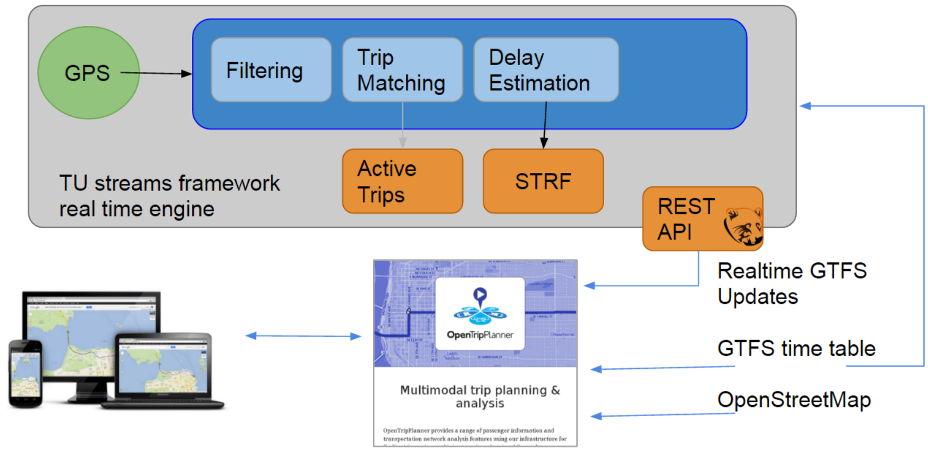


Fig. 2: Architecture of our proposed trip planner system. The real-time processing of the GPS streams detects delays of the vehicles, assigns them to trips and estimates real-time predictions of future delays. By a REST server these delays are handed in GTFS-realtime format to the OpenTripPlanner server. The user triggers a trip query with his/her browser and during trip computation real-time predictions are incorporated. Best viewed in color.

functionality. Furthermore it seamlessly compiles to Apache Kafka or Flink and thus can be integrated in state-of-the-art distributed real-time architectures.

The steps for delay estimation from the GPS stream of the transit vehicle locations are:

- Data cleaning: Removal of duplicates and noisy GPS recordings.
- Plausibility Test: Test whether the recorded GPS location is plausible given previous recordings.
- Trip Matching: Match the position of the vehicle to a trip and line of the transit graph.
- Delay Estimation: Estimate current delay of the vehicles using the assigned trip and line information, and matching it to the schedule.
- Delay Prediction: Compute estimates of future delays given the training data and past and current delay.

The latter predictions are served to OpenTripPlanner in GTFS realtime format via a Representational State Transfer (REST) webserver interface.

4 Tram Delay Prediction with STRF

The preliminary analysis of tram location data compared with schedule data, confirms the findings of [17], namely that:

– departure time not matching schedule time can be identified, but has to be analysed carefully, taking into account limited certainty of departure time estimation,
– still, noticeable number of early and late departure events can be observed in the data,
– tram delays and early departures significantly vary based on the time of the day and tram line.

This provides basis for the prediction of tram delays. Moreover, since not on time departures happen, situation-aware trip planning as an alternative to static route planning is fully justified.

In our approach, we assume that the delays do not occur at random, but follow a stochastic process. Thus, we may introduce random variables for the delay of a particular line, particular ride, and specific station. Graphical models provide an intuitive way to represent dependencies among random variables in a network structure. Thus, we model the (previously in real-time detected) tram delay by a probabilistic graphical model. Some of its random variables are related, these relations are noted by edges. In this probabilistic graphical model, we may apply observations as evidence and use loopy belief propagation to gain an estimate of the maximum a-posteriori probability. In our model, we differentiate the random variables in time and space: spatially we connect the random variables along a trip of a tram with edges, temporally we introduce one layer of this spatial structure for every ride of the line and connect edges to adjacent stations. Thus one vertex in the graph holds the delay of the corresponding ride at the corresponding stop, and one layer in the spatio-temporal random field represents the delay of one ride. The so built spatio-temporal random field not only uses discrete space and time but also discrete random variables. We distinguish these five states:

1. more than 5 minutes too early
2. 1 to five minutes too early
3. in time
4. 1 to 4 minutes belated
5. more than 4 minutes belated

When a tram passes a stop, the time of the delay is detected, and the corresponding node is set to its observed value. Afterwards, the maxprod-algorithm is applied to estimate a maximum a-posteriori (MAP) configuration.

In order to model the delay of the public transit vehicles as measured by the GPS stream, a Spatio-Temporal Random Field is constructed. The intuition behind STRF is based on sequential probabilistic graphical models, also known as linear chains, which are popular in the natural language processing community. There, consecutive words or corresponding word features are connected to a sequence of labels that reflects an underlying domain of interest like entities or part of speech tags. If a sensor network, represented by a spatial graph $G_0 = (V_0, E_0)$, is considered that generates measurements over space and time, it is appealing to identify the joint measurement of all sensors with a single word in a sentence and connect those structures to form a temporal chain

$G_1 - G_2 - \cdots - G_T$. Each part $G_t = (V_t, E_t)$ of the temporal chain replicates the given *spatial graph* G_0, which represents the underlying physical placement of sensors, i.e., the spatial structure of random variables that does not change over time. The parts are connected by a set of spatio-temporal edges $E_{t-1;t} \subset V_{t-1} \times V_t$ for $t = 2, \ldots, T$ and $E_{0;1} = \emptyset$, that represent dependencies between adjacent snapshot graphs G_{t-1} and G_t, assuming a Markov property among snapshots, so that $E_{t;t+h} = \emptyset$ whenever $h > 1$ for any t. The resulting spatio-temporal graph G, consists of the snapshot graphs G_t stacked in order for time frames $t = 1, 2, \ldots, T$ and the temporal edges connecting them: $G := (V, E)$ for $V := \cup_{t=1}^{T} V_t$ and $E := \cup_{t=1}^{T} \{E_t \cup E_{t-1;t}\}$.

Finally, G is used to induce a generative probabilistic graphical model that allows us to predict (an approximation to) each random variables MAP state as well as the corresponding marginal probabilities. The full joint probability mass function is given by

$$p_{\boldsymbol{\theta}}(\boldsymbol{X} = \boldsymbol{x}) = \frac{1}{\Psi(\boldsymbol{\theta})} \prod_{v \in V} \psi_v(\boldsymbol{x}) \prod_{(v,w) \in E} \psi_{(v,w)}(\boldsymbol{x}).$$

Here, $\boldsymbol{X}$ represents the random state of all sensors at all T points in time and $\boldsymbol{x}$ is a particular assignment to $\boldsymbol{X}$. It is assumed that each sensor emits a discrete value from a finite set $\mathcal{X}$. By construction, a single vertex v corresponds to a single stop s at a fixed point in time t. The potential function of an STRF has a special form that obeys the smooth temporal dynamics inherent in spatio-temporal data.

$$\psi_v(\boldsymbol{x}) = \psi_{s(t)}(\boldsymbol{x}) = \exp \left\langle \sum_{i=1}^{t} \frac{1}{t-i+1} \boldsymbol{Z}_{s,i}, \phi_{s(t)}(\boldsymbol{x}) \right\rangle$$

The STRF is therefore parametrized by the vectors $\boldsymbol{Z}_{s,i}$ that store one weight for each of the $|\mathcal{X}|$ possible values for each stop s and point in time $1 \leq i \leq T$. The function $\phi_{s(t)}$ generates an indicator vector that contains exactly one 1 at the position of the state that is assigned to stop s at time t in $\boldsymbol{x}$ and zero otherwise. For a given data set, the parameters $\boldsymbol{Z}$ are fitted by regularized maximum-likelihood estimation.

As soon as the parameters are learned from the data, predictions can be computed via MAP estimation,

$$\hat{\boldsymbol{x}} = \arg \max_{\boldsymbol{x}_{V \setminus U} \in \mathcal{X}} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{V \setminus U} \mid \boldsymbol{x}_U), \tag{1}$$

where $U \subset V$ is a set of spatio-temporal vertices with known values. The nodes in U are termed observed nodes. Notice that $U = \emptyset$ is a perfectly valid choice that yields the most probable state for each node, given no observed nodes. To compute this quantity, the sum-product algorithm [12] is applied, often referred to as loopy belief propagation (LBP). Although LBP computes only approximate marginals and therefore MAP estimation by LBP may not be perfect [10], it suffices our purpose.

5 Experiments

In our experiments, we use real-time GPS traces of trams in Warsaw, Poland[3], and predict current tram delays. The street network, we use, originates from OpenStreetMap, the tram schedule (in standardized GTFS format) was generated manually. As stated in previous section, we build one STRF model for every line, the stations of one trip form the spatial graph and each trip generates a temporal extrusion of the graph. Thus, a random variable is generated for the delay of a tram at every stop. The dependencies among these variables are modeled as stated in previous section. The data was trained with data recorded from June 13th till June 17th.

We apply the model to data on July 4th, 2016. In Figure 3, we plot an example query without incorporation of our real-time predictions and, beneath, with a proposed trip. The figure highlights that our approach utilizes the prediction-as-a-service and suggests trips with different tram lines or initial walking directions based on predicted delays.

A comparison of our method with the queuing model presented in [6] for line 15 can be seen in Table 1 and Table 2. As can be seen, the related approach performs worse in four of five classes, highest improvement of our method is in class two. Only in class three our method performs slightly worse.

The poor performance in the delay prediction task of both methods seems to highlight some challenges with the data we used. Possible problem could be that the tram schedule originates from a different period than the GPS data and the schedule is outdated. If so, there should be a systematic at which time and space our method performs well and worse and it should not be at random.

Therefore, we utilize the visual approach presented in [17] and inspect in more detail the accuracy of our predictions in Figure 4. Every line corresponds to one trip and every column to one stop. The vertical axis denotes the line per day and the horizontal axis the stop per trip. The patterns that are visual in the figure, e.g. high accuracy in the beginning of the day or at the end of a trip, justify our assumption that the model accuracy does not change arbitrary but depends on the schedule. Incorporation of up-to-date tram schedules is, thus, a major point for future work.

---

[3] Data was provided via `https://api.um.warszawa.pl`.

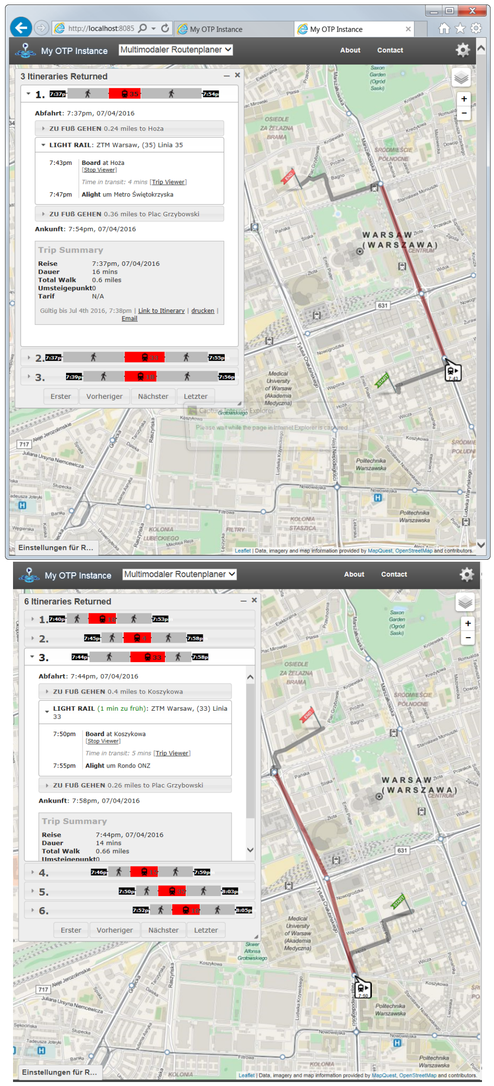


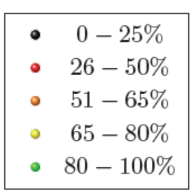
Fig. 3: Results for same start and goal. Top: without incorporation of real-time predictions, Bottom: Real-time predictions are incorporated and suggested trip avoids line 33. Best viewed in color.

Table 1: Confusion matrix of the prediction results for line 15 using our approach. Horizontally predicted classes (P) vertically True classes (T). In the end precision (Prec.) and recall (Rec.).

T \ P	1	2	3	4	5	Prec.
1	601	188	40	0	20	0.71
2	40	979	385	5	30	0.68
3	462	948	3307	40	803	0.59
4	180	128	320	197	432	0.15
5	426	252	479	120	1510	0.54
Rec.	0.35	0.39	0.72	0.54	0.54	

Table 2: Confusion matrix of the prediction results for line 15 using queueing approach by [6]. Horizontally predicted classes (P) vertically True classes (T). In the end precision (Prec.) and recall (Rec.).

T \ P	1	2	3	4	5	Prec.
1	721	9	476	1	57	0.57
2	8	28	55	0	1	0.3
3	768	69	6608	43	508	0.82
4	1	6	43	5	2	0.09
5	93	3	771	0	350	0.29
Rec.	0.45	0.24	0.83	0.1	0.38	



Legend for accuracy plot in Figure 4.

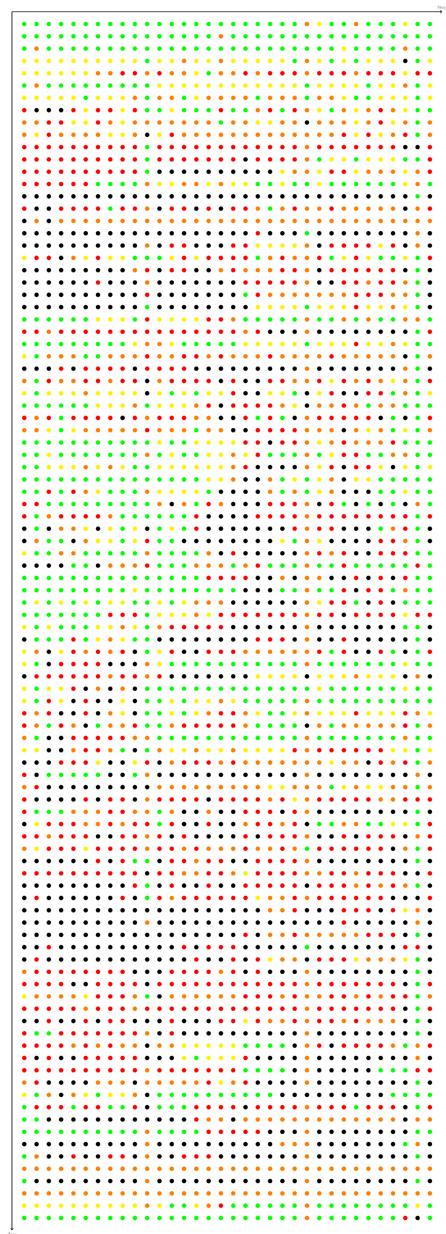


Fig. 4: Accuracy per random variable in the STRF for line 15. Depicts horizontally stop per trip and vertically trip per day, compare [17]. Find the legend next to the figure. Best viewed in color.

6 Discussion

In this work we presented a novel approach to incorporate real-time delay predictions in a multi-modal trip planner. The achieved model incorporates the predictions and generates situation-aware trips which allow for informed travel plan decisions within a smart city. These decisions can be a situation-aware initial walking direction, a situation-aware transfer from one line to another, or a different tram connection. We highlighted usability of our approach in the city of Warsaw, Poland. For real-world application the dynamic multi-modal routing has to become more efficient to handle thousands of route queries a day. A possible solution would be the incorporation of dynamic transfer pattern [2]. We studied this direction in [14]. Another important task is the combination with other modes of transportation and their predictions: vehicular traffic jams, availability of bike rentals or parking lots. In this direction it will be important to analyse how the modes of transportation interact with each others, e.g. a tram or bus is stuck in a vehicular traffic jam.

Acknowledgements. This research received funding under the Horizon 2020 programme, grant number 688380 VaVeL - Variety, Veracity, VaLue: Handling the Multiplicity of Urban Sensors. We gratefully thank Nico Piatkowski for supply of his STRF library, support and discussion.

References

1. Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F.: Route planning in transportation networks. arXiv preprint arXiv:1504.05140 (2015)
2. Bast, H., Sternisko, J., Storandt, S.: Delay-robustness of transfer patterns in public transportation route planning. In: ATMOS-13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems-2013. vol. 33, pp. 42–54. Schloss Dagstuhl?Leibniz-Zentrum fuer Informatik (2013)
3. Bockermann, C., Blom, H.: The streams framework. Tech. Rep. 5, TU Dortmund University (12 2012), `http://jwall.org/streams/tr.pdf[Lastaccessed:28November2013]`
4. Dibbelt, J., Pajor, T., Strasser, B., Wagner, D.: Intriguingly simple and fast transit routing. In: International Symposium on Experimental Algorithms. pp. 43–54. Springer (2013)
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik 1(1), 269–271 (1959)
6. Gal, A., Mandelbaum, A., Schnitzler, F., Senderovich, A., Weidlich, M.: Traveling time prediction in scheduled transportation with journey segments. Information Systems (2015)
7. Geisberger, R., Sanders, P., Schultes, D., Delling, D.: Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In: International Workshop on Experimental and Efficient Algorithms. pp. 319–333. Springer (2008)
8. Goerigk, M., Knoth, M., Müller-Hannemann, M., Schmidt, M., Schöbel, A.: The price of robustness in timetable information. In: OASIcs-OpenAccess Series in Informatics. vol. 20. Schloss Dagstuhl-Leibniz-Zentrum für Informatik (2011)

9. Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE transactions on Systems Science and Cybernetics 4(2), 100–107 (1968)
10. Heinemann, U., Globerson, A.: What cannot be learned with bethe approximations. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence. Barcelona, Spain (2011)
11. Higgins, A., Kozan, E.: Modeling train delays in urban networks. Transportation Science 32(4), 346–357 (1998)
12. Kschischang, F.R., Frey, B.J., Loeliger, H.A.: Factor graphs and the sum-product algorithm. IEEE Transactions on Information Theory 47(2), 498–519 (2001)
13. Liebig, T., Kemloh Wagoum, A.U.: Modelling microscopic pedestrian mobility using bluetooth. In: ICAART. pp. 270–275. SciTePress (2012)
14. Liebig, T., Peter, S., Grzenda, M., Junosza-Szaniawski, K.: Dynamic Transfer Patterns for Fast Multi-modal Route Planning, pp. 223–236. Springer International Publishing, Cham (2017), http://dx.doi.org/10.1007/978-3-319-56759-4_13
15. Liebig, T., Piatkowski, N., Bockermann, C., Morik, K.: Dynamic route planning with real-time traffic predictions. Information Systems 64, 258–265 (2017)
16. Liebig, T., Xu, Z., May, M.: Incorporating mobility patterns in pedestrian quantity estimation and sensor placement. In: Citizen in Sensor Networks, pp. 67–80. Springer Berlin Heidelberg (2013)
17. Mazimpaka, J.D., Timpf, S.: A visual and computational analysis approach for exploring significant locations and time periods along a bus route. In: Proceedings of the 9th ACM SIGSPATIAL International Workshop on Computational Transportation Science. pp. 43–48. ACM (2016)
18. Müller-Hannemann, M., Schnee, M.: Efficient timetable information in the presence of delays. In: Robust and Online Large-Scale Optimization, pp. 249–272. Springer (2009)
19. Niu, X., Zhu, Y., Cao, Q., Zhang, X., Xie, W., Zheng, K.: An online-traffic-prediction based route finding mechanism for smart city. International Journal of Distributed Sensor Networks 2015, 18 (2015)
20. Schnitzler, F., Artikis, A., Weidlich, M., Boutsis, I., Liebig, T., Piatkowski, N., Bockermann, C., Morik, K., Kalogeraki, V., Marecek, J., Gal, A., Mannor, S., Kinane, D., Gunopulos, D.: Heterogeneous stream processing and crowdsourcing for traffic monitoring: Highlights. In: Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, vol. 8726, pp. 520–523. Springer Berlin Heidelberg (2014), http://dx.doi.org/10.1007/978-3-662-44845-8_49
21. Schnitzler, F., Liebig, T., Mannor, S., Souto, G., Bothe, S., Stange, H.: Heterogeneous stream processing for disaster detection and alarming. In: IEEE International Conference on Big Data. pp. 914–923. IEEE Press (2014)
22. Souto, G., Liebig, T.: On event detection from spatial time series for urban traffic applications. In: Michaelis, S., Piatkowski, N., Stolpe, M. (eds.) Solving Large Scale Learning Tasks: Challenges and Algorithms, vol. 9580, pp. 217–230. Springer International Publishing (2016)
23. Utsch, P., Liebig, T.: Monitoring microscopic pedestrian mobility using bluetooth. In: Intelligent Environments (IE), 2012 8th International Conference on. pp. 173–177. IEEE (2012)
24. Zygouras, N., Zacheilas, N., Kalogeraki, V., Kinane, D., Gunopulos, D.: Insights on a scalable and dynamic traffic management system. In: EDBT. pp. 653–664 (2015)