

Deviations Prediction in Timetables Based on AVL Data

Zbyněk Jiráček, Vladislav Martínek, and Miroslav Čermák

Dept. of Software Engineering,
Charles University in Prague,
Prague, Czech Republic
JBI@seznam.cz,
martinek@ksi.mff.cuni.cz,
cermak@ksi.mff.cuni.cz

Abstract: Relevant path planning using public transportation is limited by reliability of the transportation network. In some cases it turns out that we can plan paths with respect to expected delays and hereby improve reliability of the resulting path. In our work we focus on prediction of the delays in public transportation systems. For this purpose we use data from vehicle tracking systems used by transit operators - known as the AVL data.

We compare statistic methods to methods of artificial intelligence using data from Prague trams tracking system. We discovered that in some cases the neural networks show better results than the statistic methods. In contrast, sometimes even simple statistical methods give as good results as those provided by the neural networks.

1 Introduction

Many people need to travel almost every day. When they do so, they usually have two options - to use individual transportation means (e.g. a car) or to use public transportation. When making this decision, many aspects are taken into account. One of the important aspects besides price and duration of the journey is reliability.

Especially in larger cities, lot of money is often spent to support public transportation systems, including reliability improvements - mostly by building underground lines and by segregation of trams and buses from individual traffic.

An alternative way of improving reliability is by providing useful information. Given that we will never be able to ensure that the system is 100% reliable, we can soften the consequences of traffic irregularities by warning the passengers. In more advanced case we can improve path-planning systems in such way that they prefer more reliable paths. The main advantage of this approach to dealing with irregularities is that it is relatively simple and cheap. Transit operators usually already collect tracking data from vehicles. These data are known as Automatic vehicle location (AVL) data. AVL data give us information about positions of vehicles, typically in real-time.

In this article we find and compare approaches how to interpret the information from AVL data and how to predict future development of delays of the public transportation vehicles in real-time. We assume that the resulting information can be transmitted to passengers for instance via a mobile application. Since the number of people who

possess a smart mobile device is increasing every year, this also means an increase of potential users of this kind of application. Additionally, transit operators can equip stops by information systems presenting this information to passengers.

We compare three methods: statistical, regression and neural networks. The statistical methods are expected to be computationally the least complex while the neural networks should provide the best results.

We also define and compare static and dynamic prediction. By static prediction we mean models that use only past data and do not have access to the information in real-time. Therefore, they can predict only expectable deviations that happen repeatedly.

Unlike static prediction, dynamic prediction does not use only past data, but takes real-time information into account as well. This allows recognition of unexpected problems in the transit network. On the other hand, real-time data have only short-term validity and passengers can use information from dynamic prediction system only when they are about to travel somewhere or while they are already travelling.

The structure of the article is as follows: The first section introduces the problem and the goal. The second section shows previous and related work. In the third section we analyse the problem in detail and in the fourth section we evaluate the selected methods. In the fifth section we compare and discuss the results. The sixth section offers some ideas about practical usage of the results. And the seventh section concludes the article and offers future focus.

2 Related Work

There is a lot of work related to reliability of public transport. Common public transport unreliability issues are discussed by Rietveld et al. in [1]. A static prediction algorithm was already presented by Martínek and Žemlička in [2]. The algorithm corrects the timetable given by the transit operator and the result is then presented to passengers. Similar approach is mentioned by Dessouky and Randolph [3]. They treat travel time as a log-normal distributed random variable and calculate the expected travel time as the timetable time plus mean delay.

An example of a dynamic approach is described by Tien et al. in [4]. The system, though, does not predict situation, it only checks if the user’s progress corresponds to the schedule and, if not, it computes an updated schedule or a new route.

A dynamic prediction using statistical methods was used by Wall and Dailey in [5]. Jeong and Rillet compared regression methods and neural networks on a bus line in Houston, Texas [6]. According to their measurements, neural networks seem to be more accurate and more promising. More general comparison of neural networks and statistical methods in transportation is provided by Karlaftis and Vlahogianni [7]. They point out that neural networks are better at recognition of more complex non-linear relations. But a significant drawback of neural networks is that they are much less transparent than statistical methods.

Bohmova and Mihalak [8] suggest that in larger networks where each line is served with high frequency we can guide passengers only by a list of stops and lines. It means that instead of information “take line X departing at HH:MM” we tell the user to take first vehicle of line X or Y in a specific direction.

3 Our Focus

We focus mostly on transportation networks in larger cities. These networks usually have more complex structure without clear hierarchy and there are several options of getting from one location to another.

According to Rietveld at al. [1] there are two major causes of unreliability in public transport - recurrent and non-recurrent congestion. While recurrent congestion occurs every weekday at particular times and places, non-recurrent congestion is caused by unpredictable incidents. Non-recurrent congestion is relevant especially when talking about trams or trains. A smaller incident can affect more passengers since rail vehicles are typically not able to bypass the critical spot. Note that non-recurrent congestion cannot be predicted statically. However, dynamic prediction mechanisms can identify the problem since they have access to information about current situation.

Rietveld at al. also mention an obvious trade-off. Faster transport or shorter halting times will improve the scheduled travel times, but will have an adverse effect on the reliability of the service [1]. This motivates us to study the impact of the current delay on the additional future delay of the same vehicle. Sometimes when a vehicle is far behind its schedule, it creates a longer interval between this vehicle and the previous one on the same line. If the line is served with high frequency, passengers usually do not consult schedules and arrive randomly at their stops [3]. Larger interval in this situation means that the delayed vehicle must transport more passengers. More passengers cause longer dwell times spent at stops, which can lead to further delays if there is not enough spare time in the

timetable. Furthermore, the next vehicle, if it is on time, will have less passengers to serve and therefore will stay on time more easily.

We were able to locate the effect described above in the data from Prague trams tracking system we have available. Let a be a stop somewhere near the middle of a specific line and let z be the final stop of the same line. We would like to express the relationship between $D(a)$ being the delay at the stop a and $D^+(a, z) = D(z) - D(a)$ which is the additional delay at the stop z . We divided the past observations into clusters by $D(a)$ value; each cluster contains connections with delay from k to $k + 1$ minutes at the stop a . Then we expressed the average additional delay for each of these clusters. If the delays on the way were independent, these values should be similar. But figure 1 shows there most probably is a relationship. While on the line 9b the drivers are usually capable of reducing the delay, the line 22a shows that the more delayed a vehicle is, the even more delayed it usually is on the rest of its way to the final stop.

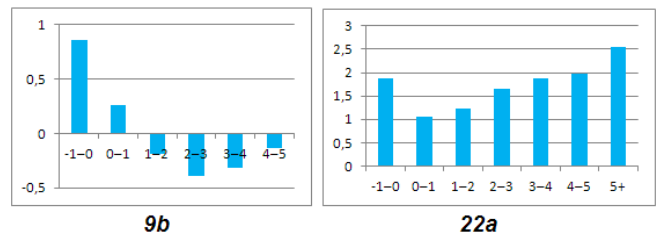


Figure 1: The average additional delay on arrival to the final stop for lines 9b (left), and 22a (right).

4 Used Methods

We have data from Prague trams tracking system from March and April 2008. When a tram serves a stop, it also sends a message to the tracking system where this information is stored. Therefore we don’t know the exact position of the tram in every moment; we have only information about the last stop the tram stopped at together with the associated time.

For objective evaluation of the methods we divided the data into learning set and test set. The learning set is about two times larger and is used as input when creating a model. The test set is then used to evaluate the model. The learning set contains data from March and the beginning of April, while the test set contains the rest of the data until the end of April. This way we simulate the real situation - that we create the model on the past data and use the model for predictions.

For evaluation of the methods we calculate the following metrics. In these metrics, an error is the difference between the predicted and real arrival.

- The average absolute error

- Median absolute error
- Mean absolute percentage error (the absolute error divided by the actual travel time)
- 95% confidence interval of the absolute error
- Percent of connections with absolute error under 60 seconds

For a particular tram currently located at a certain stop that we call the initial stop, our task is to predict the times of arrival to the following stops on its path. For simplification we have chosen three lines (parts of the lines, respectively) with different characteristics. We also chose for each line one initial stop a and one target stop b instead of predicting arrivals to all remaining stops on the line. The line parts we have chosen as the test subjects are described in the following paragraphs.

Line 9b in the selected part is completely segregated from the individual transport and there are no traffic lights on its track. Therefore it is rarely delayed and if so, the drivers are usually capable of decreasing the delay on the way (see figure 1).

Line 22a in the selected part has some intersections equipped with traffic lights on its way. This makes the vehicle movement more unpredictable, but does not cause major delays.

Line 22b in the selected part is not segregated from individual transport and sometimes it suffers from recurrent and non-recurrent congestions much more than the previous two lines.

For all the three lines, the travel time from the stop a to the stop b is approx. 12–15 minutes.

We used three methods of prediction on these lines in Matlab. Simple statistical method, neural networks and regression.

In the following sections we use this notation:

L : The learning set of past observations on a particular line. An observation is a set of times and delays for each stop on the line. An observation corresponds to a single connection performed by a tram vehicle in the past.

a : The initial stop.

b : The target stop.

$D(c, s)$: The delay of a specific tram connection c at the stop s .

$D^+(c, a, b)$: The additional delay of a specific tram connection c between stops a and b . This value is equal to $D(c, b) - D(c, a)$.

4.1 Statistical Methods

The most straightforward solution is to calculate the average additional delay $\overline{D^+}(a, b)$ between stops a and b in the following way:

$$\overline{D^+}(a, b) := \frac{1}{|L|} \sum_{c \in L} D^+(c, a, b) \quad (1)$$

Now in the present situation we have a vehicle v currently located at the stop a . Therefore we know $D(v, a)$. We want to predict $D(v, b)$. In the following formula, let $\tilde{D}(v, b)$ be the prediction of $D(v, b)$.

$$\tilde{D}(v, b) := D(v, a) + \overline{D^+}(a, b) \quad (2)$$

Since we use the current delay to predict the future delay, this is a dynamic prediction algorithm. We can compare it with a static version, which corresponds to the approach provided by Martínek and Žemlička in [2]. In the static version we calculate the expected delay without usage of the value $D(v, a)$, which we don't know in the moment of the calculation:

$$\tilde{D}(b) := \frac{1}{|L|} \sum_{c \in L} D(c, b) \quad (3)$$

In the formula above we use average delay at the target stop instead of the average additional delay. Note that since this is a static calculation, it does not depend on the concrete vehicle v . Finally, we compare the static and dynamic calculations using our dataset. The table 1 shows the average absolute errors for both static and dynamic prediction algorithms.

Clustering Similarly as in [2] we can divide the data to workdays and weekends and cluster the average values by hours to get finer resolution, since the delays in morning hours may differ from those at evenings. This means that instead of one $\overline{D^+}(v, b)$ value we have 2×24 values for each hour for workdays and weekdays separately. In the equation 2 we use one of the 48 values based on the current time and the day of week. The table 1 shows the improvement in average prediction error.

Method	Average absolute error		
	9b	22a	22b
Static non-clustered	64.8 s	90.8 s	170.5 s
Static clustered	62.6 s	90.5 s	150.1 s
Dynamic non-clustered	37.1 s	51.2 s	113.7 s
Dynamic clustered	32.5 s	47.0 s	97.2 s

Table 1: Average absolute error for static/dynamic clustered/non-clustered statistic prediction

4.2 Neural Networks

Neural networks are commonly used in transportation research (see [6] and [7]). Their main advantage is that they can handle multi-dimensional data and are capable of recognition of non-linear relationships. The main disadvantage lies in the lack of transparency. It is usually very hard to explain the results calculated by the neural networks.

We learned the neural networks on the past data using the Levenberg-Marquardt method with 10 neurons in one

hidden layer, which showed the best performance and accuracy in our tests. More information about structure and learning process of the neural networks can be found in literature [9]. We used neural networks toolbox in Matlab. Some of its advantages are a built-in protection against overfitting and automatic normalization of the input.

For each past connection observation from the learning set $c \in L$ we created an input vector. The input vectors consisted of the time, day of week and delays at each stop from the starting point of the line to the initial stop a . The network had only one output value - the prediction of the delay at the stop b^1 . The first results are shown in the table 2.

Statistics on lines	9b	22a	22b
Average absolute error	29.8 s	45.6 s	96.5 s
Median absolute error	22.2 s	36.0 s	59.2 s
Mean percentage error	5.36 %	8.81 %	14.51 %

Table 2: Prediction precision using neural networks

As the results are very similar to those provided by the simple statistical methods (see table 1), we decided to make some improvements to the neural network.

Input When changing the structure of the input vectors, we found that the network does not use delays from the previous stations before the station a . Additionally, the weekday information could be simplified to a boolean value “is-workday”.

As a result, only three-element vectors were used as the input: The time, workday boolean, and the delay at the stop a , while the results did not change.

Topology We have tested many different topologies of the network. It turns out that a neural network with approx. 10 neurons in one hidden layer is sufficient. In rare cases the neural network failed to learn, which can be improved by adding one more layer. Adding more neurons and layers only slowed down the learning process, but did not improve the results. Changing the learning method did not bring any improvements as well.

Clustering Similarly as in statistical methods we tried to divide the data, at first only into two groups - workday and weekends. Then we learned two neural network models and for each input we used the appropriate model. We found that this approach only worsens the previous results. Later, we found that Jeong and Rillet in [6] have observed the same effect.

Current situation Until now we used only data about the particular connection when constructing a single input

vector. The neural network treats the input vectors independently and therefore when it is asked for an output, it can use only the data specified in the input vector. That means the network did not use any other information, for instance about status of the previous vehicles on the same line.

However, we would like the network to use more information about current situation. In order to do that we need to extend the input vector and encode the information into it.

We have decided that we try to improve the results by adding information about a few previous vehicles on the same line. Question is, how to express this information in a form of a vector that a neural network would be able to understand. The fact that we do not know the exact positions of the trams, but only the last served stop, also needs to be taken into account.

Given the limitations above we extended the input vector by two values: number of trams of the given line currently located between stops a and b , and travel time from a to b of the last tram on the given line that has reached the stop b . This improved the results for line 22b by approx. 25 %, but did not bring any significant changes of the results for lines 9b and 22a. Table 3 shows how the absolute prediction error has changed.

Statistics on lines	9b	22a	22b
Average absolute error	32.7 s	45.8 s	72.6 s
Median absolute error	26.1 s	37.0 s	49.1 s
Mean percentage error	5.91 %	8.76 %	11.41 %

Table 3: Prediction precision using neural networks with more inputs

We explain these results by the differences between the lines. As the lines 9b and 22a are not directly influenced by other types of transport, their delays are more random. On the contrary, the line 22b is highly influenced by current traffic situation in the area, which usually does not dramatically change within just a few minutes. Therefore if a tram on the line 22b is delayed, it is probably caused by the traffic congestions and it is also probable that the next tram on this line will be delayed too.

Further improvements The input vectors now contain more information about current situation, yet the data presented are still very limited. The network uses the information about the last tram that has passed the stop b . This might still not be optimal.

If the distance between stops a and b is for example 15 minutes, we use information about a tram which is 15 minutes ahead. And this 15 minutes is a time long enough for the situation to change and therefore the prediction may be based on obsolete information.

Moreover, as the trams send their location to the system only at stops, this can cause problems in case of an accident. If a tram is extremely delayed or stopped on its

¹Actually, the output could be easily widened to produce one prediction value for each stop; we use only one-dimensional output for clarity.

way, we are not informed about that. The only way how to assume this is by the fact that the tram has not arrived to the stop b for a long time. This, again, slows the reaction of the network, since it takes some time before a tram becomes late enough to be suspicious.

What could improve the results is the knowledge of the exact position of the tram in real-time (or in reasonable intervals). But given the nature of the system used in Prague this is rather unrealistic.

The only option left is a better usage of the data presented. For instance the input vector could contain information about delays of trams at stops between a and b , which it does not now. Or it could contain information about trams from other lines that share a part of their path with the current line. Nevertheless, it is necessary to present the values in such way that the neural network will be able to interpret the data. We believe there may be a chance of further improvements regarding to this matter, though we were not able to devise an input form that would prove that.

4.3 Regression

The similarities between the results of statistical processing and neural networks encouraged us to try one more method – regression. Regression should provide better results than simple statistics, which in some cases gave as good results as the neural networks.

Inspired by the input of the neural networks, we used the following linear equation for the regression:

$$D(v, b) = k_1 T(v) + k_2 W(v) + k_3 D(v, a) + k_4 NP + k_5 D(w, b) + k_6$$

where:

$D(v, b)$ is the delay of the vehicle in the target stop,
 $T(v)$ is the time of departure of the vehicle,
 $W(v)$ is 1 for workdays, or 0 for weekends,
 $D(v, a)$ is the delay of the vehicle in the initial stop,
 NP is the number of vehicles on the same line currently located between stops a and b ,
 $D(w, b)$ is the delay of the last vehicle on the same line that has passed the stop b ,
 k_i are coefficients we want to solve by the regression.

Note that the inputs to this equation are the same we used for the neural networks in section 4.2.

The results are summarized in the following table:

Statistics on lines	9b	22a	22b
Average absolute error	32.0 s	48.0 s	73.9 s
Median absolute error	25.6 s	38.2 s	52.1 s
Mean percentage error	5.76 %	9.11 %	11.81 %

Table 4: Prediction precision using linear regression.

The comparison to the other methods is offered in the section 5.1.

4.4 Improvements

Similarly as we did with the previous methods we tried to apply some improvements. First, adding higher degrees of the time and delay input variables did not change results significantly. Neither did clustering of the results. It may be possible that there is some combination of the input variables that could lead to better results, but we believe it is unlikely.

5 Comparison

In this section we would like to compare the results from the previous sections.

5.1 Used Methods

First we compare the used methods. The table 5 shows the final results.

Method	Average absolute error		
	9b	22a	22b
Statistical processing	32.5 s	47.0 s	97.2 s
Neural networks	29.8 s	45.6 s	72.6 s
Regression	32.0 s	48.0 s	73.9 s

Table 5: Best average absolute error for particular prediction methods.

The results indicate that on lines 9b and 22a all the methods present similar predictions. We think that this is caused by relatively good punctuality rate of these two lines. Average delay of the line 9b at the target stop is 38 seconds, for the line 22a it's 94 seconds. The average delay for the line 22a is higher, but the delays on this line are probably caused mostly by the three intersections equipped with traffic lights, which generate unpredictable deviations. Together they can hold a tram for approx. 180 seconds in the worst case.

The results on lines 9b and 22a suggest that in the traffic network where there are only small delays, or the delays are caused mostly by unpredictable factors, simple statistical methods are most suitable. Implementation of linear regression or even neural networks is far more complex and most probably does not bring any improvements in these situations.

Regarding the line 23b, neural networks, together with linear regression outperformed the statistical prediction. This is mostly given by the fact that the statistical methods we used were not able to process many-dimensional input. The neural networks and regression have become more precise by adding information about previous vehicles to the input, which we cannot as simply add to statistical methods too. Before we added this data to the input of the neural networks and regression, the results were similar for the line 23b too.

The results show no significant difference between the precision of neural networks and linear regression. This is a little surprising as we expected the neural networks to be capable of discovering more complex non-linear relationships between the input and output data.

5.2 Static vs. Dynamic Prediction

In this article we also wanted to compare the static and dynamic methods. It is clear that the dynamic methods should provide more accurate predictions; the purpose of this comparison is more to express the improvement that the dynamic prediction methods can offer.

To simulate the static environment we used the same methods: statistics, neural networks, and regression. The only difference is that static methods do not know the actual timetable deviations and therefore do not have the $D(v,a)$ value on the input. The result is that the static methods must predict the delay $D(v,b)$ using only the time and the day of week (based on the past observations).

First we compared the static versions of the used methods to each other. The result is that in the static environment all the three methods give almost the same results.

Then we compared the static and dynamic methods. The table 6 compares the best static method with the best dynamic method results.

Statistics for lines		9b	22a	22b
Mean abs. err.	Stat	63 s	90 s	150 s
	Dyn	30 s	46 s	73 s
Median abs. err.	Stat	47 s	67 s	104 s
	Dyn	22 s	36 s	49 s
Mean perc. err.	Stat	11.3 %	17.2 %	24.2 %
	Dyn	5.4 %	8.8 %	11.4 %

Table 6: Comparison of static (Stat) and dynamic (Dyn) prediction methods.

6 Possible Usage

6.1 Information Systems

The results that the algorithms present could be used in public transport information systems. The simplest usage of the data is a direct presentation of the results to passengers via mobile phones or information systems at stops. These systems already exist in many cities, they normally list departures from a particular stop, ordered by the departure time. These systems typically show only static timetable data, sometimes together with current delays.

But the passengers are not actually interested in the current delay; the information is provided to them so that they can infer the real departure time, which is inherently predicted as the timetable departure + the delay. However, we could provide the users with the predicted departure, which should be more accurate.

We decided to compare those values. In the table 7, “Inherent prediction” is a simple algorithm that always predicts the future delay as the same value, as the current delay - formally $\bar{D}(v,b) := D(v,a)$. In fact, this is the simplest possible dynamic prediction algorithm. For reference we also added “No prediction” algorithm which simply uses the value from timetable and assumes zero delay. This represents the simplest possible static prediction algorithm.

Method	Average absolute error		
	9b	22a	22b
No prediction	64 s	114 s	258 s
Inherent prediction	46 s	58 s	127 s
Statistical processing	33 s	47 s	97 s
Neural networks	30 s	46 s	73 s

Table 7: Comparison of simple delay estimates and prediction algorithms.

The numbers show that using the prediction algorithms, we can reduce the departure prediction errors. With usage of more advanced methods like neural networks or regression, the improvement can be even greater.

6.2 Navigation

The prediction data can be also used in public transport connection search engines. These applications typically search only in timetables and do not reflect current situation. If the systems used predicted departures and arrivals, they could possibly be able to find faster and more reliable connections. Especially when the user is searching for the fastest connection “right now”, we could use the benefits of the dynamic predictions.

The most complex systems are public transport navigation systems. These applications are often capable of dealing with delays at least in a simple way. Such a system was already implemented in Boston [4]. Adding a prediction unit to such systems might improve their reliability.

Example Situation Mike is currently at a stop a and needs to get to stop b to catch a train. There are two lines, 18 and 22, that connect the stops a and b , each of them goes a different way. Mike knows that the line 22 has higher probability to be delayed between the stops a and b . Both lines can also be delayed on their way to the stop a . A tram 22 is approaching the stop a , while the tram 18 is scheduled a minute later. What should Mike do? Should he board the approaching tram and risk the possible delay on the way? Or would it be better to wait for the tram 18 and risk that it will arrive late?

Solution This situation can be solved by the prediction algorithms. If Mike had access to information from such system, he would know that because of a bad traffic situation the trams on line 22 are predicted to be delayed by

5–10 minutes along the route to the stop *b*. He would also know that the tram 18 scheduled a minute later is on time. This would help him to decide not to board the tram 22 and wait for the tram 18, which would most probably help him to get to the stop *b* on time.

7 Conclusion

We have shown that by using even simple prediction algorithms, it is possible to predict movement of the public transportation vehicles much more precisely than just by using the timetables given by the transit operator. It also turned out that for lines with only small or unpredictable delays, more complex methods like regression or neural networks are not more accurate than the basic statistical methods. Therefore, usage of regression or neural networks is reasonable only in environments with significant delays. As the neural networks are more complex, and most probably harder to implement, linear regression seems to be a good solution.

We also compared static and dynamic algorithms. The results indicate that when we have information about current situations, the predictions are up to twice as much accurate. Of course the results of the dynamic algorithms are valid only for a short period of time, as the situation changes.

7.1 Future Work

From the data we have available it turned out that trams in Prague are quite precise, with only a few exceptions. We believe this is the major cause of why the more complex prediction methods did not outperform the simple ones greatly. We think it would be interesting to test the algorithms on a network with more significant deviations too, for example on the Prague bus operation data, as the buses tend to be less precise because of lower level of segregation from individual transport. However, we do not have access to this data, so we could not test it.

We would also like to focus on further improvements in accuracy. We believe that the neural networks and maybe the regression too, have potential to give better results if they had more information on the input. The problem is how to encode all the information about the current situation into a vector of real values of a reasonable length.

In the future work we would also like to focus on the usage of the data from the prediction algorithms. We believe that presentation of this data to passengers in a user-friendly form is a relatively simple yet modern way how to make public transportation more attractive.

Acknowledgment

This work was supported by project GAUK 472313.

References

- [1] Rietveld, P., Bruinsma, F. R., van Vuuren, D. J.: Coping with unreliability in public transport chains: A case study for Netherlands, *Transport Research Part A: Policy and Practice* (2001) 539–559
- [2] Martínek, V., Žemlička, M.: Passenger Path Plan Reliability Improvement Proposal, *The Fourth International Conference on Information, Intelligence, Systems and Applications University of Piraeus, Piraeus, Greece* (2013) 242–247
- [3] Dessouky, M., Hall, R., Zhang, L., Singh, A.: Real-time Control of Buses for Schedule Coordination at a Terminal, *Transportation Research Part A: Policy and Practice* (2003) 145–164
- [4] Tien, D. N., MacDonald, T., Xu, Z.: TDplanner: Public transport planning system with real-time route updates based on service delays and location tracking, *IEEE Vehicular Technology Conference* (2011).
- [5] Wall, Z., Dailey, D. J.: An Algorithm for Predicting the Arrival Time of Mass Transit Vehicles Using Automatic Vehicle Location Data, *Transportation Research Board, 78th Annual Meeting, Washington, D. C., USA* (1999)
- [6] Jeong, R., Rillet, L. R.: Bus Arrival Time Prediction Using Artificial Neural Network Model, *IEEE Intelligent Transportation Systems Conference, Washington, D. C., USA* (2004) 988–993
- [7] Karlaftis, M. G., Vlahodianni, E. I.: Statistical methods versus neural networks in transportation research: Differences, similarities and some insights, *Transportation Research Part C* 19 (2011), 387–399
- [8] Bohmova, K., Mihalak, M., Proger, T., Sramek, R., Widmayer, P.: Robust Routing in Urban Public Transportation: How to find reliable journeys based on past observations, *13th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ser. OpenAccess Series in Informatics (OASISs), D. Frigioni and S. Stiller, Eds., vol. 33. Dagstuhl, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, (2013) 27–41. [Online]. Available: <http://drops.dagstuhl.de/opus/volltexte/2013/4242>*
- [9] Beale, R., Jackson, T.: *Neural Computing: An Introduction*, IOP Publishing, Bristol and Philadelphia (1990)