






# Machine Learning for Bus Travel Prediction

Lukasz Palys<sup>1</sup>, Maria Ganzha<sup>1</sup> <sup>(✉)</sup> , and Marcin Paprzycki<sup>2</sup> 

<sup>1</sup> Warsaw University of Technology, Warsaw, Poland

maria.ganzha@pw.edu.pl

<sup>2</sup> Systems Research Institute Polish Academy of Sciences, Warsaw, Poland

marcin.paprzycki@ibspan.waw.pl

**Abstract.** Nowadays, precise data of movements of public transport can be collected. Specifically, for each bus, geoposition can be regularly obtained and stored. In this context, an attempt to build a model to represent behavior of busses, and predict their delays, is discussed.

## 1 Introduction and Related Work

Recently, Warsaw, Poland, opened its data resources, including regularly collected, location of city buses. Based on this information an attempt was made to predict delays in public transport. This problem has been discussed in the literature. Applied methods can be divided into groups. First, *statistical methods*, such as k-NN [2, 7], regression [2], prediction based on sequence patterns [9] or time series [10], and application of Kalman filter [6]. Second, *methods based on observations*, which include the historical means approach [1, 5]. Next, *methods based on machine learning*, which include (see, also, [8]): back propagation neural network [11], radial basis function network [12], multilayer perceptron [1, 5]. The last group consists of *hybrid methods* that combine multiple algorithms into a single model [4, 13]. Let us now present selected results, while [15] contains detailed discussion.

The Pattern Sequence-based Forecasting was used in [9], to model bus transport in Chennai (India). Overall, K-means algorithm delivered best performance. Predictions have been tested on sections with low, medium and high travel time variability. For sections with large time variance, results were not “satisfactory”.

Time series approach was used in [10] to forecast bus travel time in Lviv (Ukraine). The average error, for travel time towards city center was approximately 3 min, and 2 min in the opposite direction.

In [12], authors used Radial Basis Function Neural Network (RBFNN) to model bus transport in Dalian (China). The model was trained on historical data, with additional correction of results (using Kalman Filter). Overall, the best reported mean absolute percentage error (MAPE) was at 7.59%.

Work described in [13], is based on deep learning and data fusion. Models used multiple features, including: stop ID, day of the week, time, bus speed (based on GPS), stopping time at a stop, travel time between stops. Data came from Guangzhou and Shenzhen (China) for single line in each city. Predictions

were compared with historical averages. Proposed model outperformed other approaches with MAPE of 8.43% (Gangzhou). Moreover, MAPE for the peak hours was 4–7% lower than that reported for other solutions.

In [1], dynamic model was developed, to predict bus arrival times at subsequent stops. GPS data, from Macea (Brazil), for a bus line with 35 stops, was used. Here, *Historical Average (HA)*, *Kalman Filtering* and *Artificial Neural Network (ANN)* were tried. The 3-layer perceptron achieved best MAPE (18.3%).

In [2], review of methods modeling bus transport, found in [3], was extended. Authors used: *k-NN*, *ANN*, and *Super Vector Regression (SVR)* to predict bus travel times in Trondheim (Norway). To study influence of individual time intervals, 423 different data sets were created. Depending on data set, (best) MAE varied between 61 and 86 s. Separately, additional attributes, e.g.: weather, football matches, tickets, were tried, with no visible improvement.

In [4], an ANN was used, with historical GPS data and an automatic toll collection system data. Moreover, impact of intersections with traffic lights was taken into account. Data came from Jinan (China), for a single bus line. To deal with travel time variation, a hybrid ANN (HANN) was developed, with separate subnets, trained for specific time periods, e.g. working days, weekends, peak hours. Overall, ANN and HANN were more reliable than the Kalman Filter. Moreover, HANN was better suited for short-distance prediction.

Separately, comparison of methods modeling tram travel in Warsaw (Poland), on the basis of historical GPS data, is presented in [5].

Main findings from the literature can be summarized as follows. (a) **Reported results** concern a single city. Only in one case two cities have been studied. (b) In all cases a **single bus line** was used. (c) There is **no “benchmark” data** for the problem. (d) The **main methods** that have been tried were: (i) statistical methods – k-NN, regression model, Kalman filter, (ii) historical observation methods (HA), (iii) machine learning methods – BPNN, RBFN, multilayer perceptron (MLP), and (iv) hybrid methods combining the above algorithms into one model (HANN). Among them, best results have been reported for HANN, HA, RBFN and MLP. However, it was reported that HANN requires substantially larger datasets. (e) **Quality of predictions** has been measured using: (i) mean absolute error (MAE; in seconds); (ii) mean percentage absolute error (MAPE); (iii) standard deviation (STD; in seconds). (f) The **simplest approaches** used GPS data alone. Other popular data elements were: information about sold tickets and about bus speed. Additionally, effects of non-travel events (e.g. games or weather) have been (unsuccessfully) tried. (g) **Best accuracy** was reported in [2], where MAE was 40s. However, this result was obtained for 1 to 16 stops only. For a “longer bus line” best MAE was of order 60s-70s. Finally, for long Warsaw tram lines (more than 40 stops), best MAE was at 123s.

## 2 Data, Its Preprocessing, and Experimental Setup

As a part of the project *Open data in Warsaw*<sup>1</sup>, exact location of public buses, reported in real time, is available. From there, 30 days of data reporting bus

<sup>1</sup> <https://api.um.warszawa.pl/>.

movements was harvested (total of about 10 GB of data). Data was filtered, retaining: (1) line number, (2) departure time from the last stop, (3) current percentage of distance traveled between adjacent stops, (4) time of the last GPS signal, (5) current time, (6) driving direction. Additionally, file containing timetable of buses, their routes, including list, and GPS coordinates, of stops, and departure times from each stop, was downloaded from the ZTM site<sup>2</sup>. After preprocessing, file with: line number, time, vehicle and brigade number, driving direction, number of the next stop to visit on the route, information whether the vehicle is at the stop, the percentage of distance traveled between consecutive stops, was created. All preprocessed data is available from: <https://github.com/lukaspal97/predicting-delays-in-public-transport-in-Warsaw-data>.

From available data, 29 bus lines have been selected. Based on “manual” analysis of their routes (in the context of Warsaw geography), the selected bus lines have been split into eight semi-homogeneous groups:

1. Long routes within periphery North-South: bypassing the City Center, running on the western side of the Wisła River; lines: 136, 154, 167, 187, 189.
2. Long routes within periphery West-East: bypassing the City Center, crossing the Wisła River; lines 112, 186, 523.
3. Centre-periphery: routes with one end in the City Center, running to the peripheries; not crossing the river; lines 131, 503, 504, 517, 518.
4. Long routes through Center (with ends on peripheries); lines 116, 180, 190.
5. Express: a fairly straight long lines with small number of stops (typically around 1/3 stops of “normal” lines); lines 158, 521, 182, 509.
6. Centre-Praga: short routes starting in the City Center; crossing the Wisła River; lines 111, 117, 102.
7. Short lines within peripheries: in Western Warsaw; lines 172, 191, 105.
8. Short lines within the Center: not crossing the river; lines 128, 107, 106.

Travel time distribution differs between days of the week (e.g. working days vs. weekends). Therefore, following [2], all models were trained on data from the same day of the week, from three weeks, and tested on data from the last (fourth) week. For each model, the best results are reported. In general, methods that use “extra features” were compared to these that use only GPS location. Accuracy was measured using MAE and STD.

### 3 Experimental Results and Their Analysis

Let us now summarize the experimental results<sup>3</sup>. We start with **Total Travel Time Prediction**. Here, two methods have been tried: “recursive” and “long distance”. In the *recursive* method, the model is trained on data that includes travel time to the nearest stop. Hence, estimating travel time from stop  $n$  to  $n + k$  consists of predicting  $k$ -steps using the trained model, i.e. result from the

<sup>2</sup> <https://www.wtp.waw.pl>.

<sup>3</sup> All reported models have been implemented in Python, using Keras library.

previous stop is included, as the input data, for the next prediction. The *long distance* method is based on prediction of travel time to a specific stop. Here, training dataset includes information about total travel time. The assumption was that this approach would be worse for short distances, but better for long(er) trips.

The comparison was made for bus line 523. Training data originated from: March 11, 18, 25. Testing data was from April 1. The number of records in the training set was over 1 million, and in the test set over 300,000. Both approaches used MLP with two hidden layers consisting of 6 and 24 neurons, with ReLU activation function. Results are presented in Table 1. As can be seen, for travel longer than 4 stops, the long distance method was more accurate than the recursive method. Moreover, for distances longer than 8 stops, most results were more than twice as accurate. For 1–3 stops, results of both methods were comparable.

**Table 1.** Prediction of total travel time (bus line 523)

Using the recursive method								
Distance	Time							
	7:00–10:00		10:00–14:00		14:00–19:00		19:00–23:00	
	MAE	STD	MAE	STD	MAE	STD	MAE	STD
1–3	58.24 s	62.76 s	60.01 s	79.94 s	59.51 s	75.74 s	47.02 s	54.01 s
4–8	125.52 s	129.44 s	130.89 s	141.44 s	131.74 s	124.96 s	92.10 s	88.65 s
9–15	211.36 s	165.19 s	214.10 s	179.41 s	240.66 s	170.14 s	156.74 s	121.10 s
16–20	294.83 s	188.44 s	310.42 s	219.57 s	344.49 s	201.74 s	222.66 s	137.57 s
21–27	360.19 s	200.56 s	398.48 s	266.61 s	407.73 s	217.13 s	277.52 s	154.40 s
Using long distance method								
1–3	56.72 s	66.06 s	59.97 s	69.67 s	59.07 s	64.44 s	54.82 s	51.11 s
4–8	81.07 s	97.31 s	85.59 s	106.85 s	78.52 s	87.18 s	66.44 s	68.93 s
9–15	107.01 s	121.46 s	101.93 s	129.62 s	105.89 s	113.19 s	85.65 s	87.78 s
16–20	130.41 s	144.66 s	140.38 s	170.57 s	130.76 s	138.80 s	113.58 s	106.45 s
21–27	153.70 s	161.04 s	170.72 s	205.03 s	155.09 s	152.63 s	172.30 s	135.29 s

**Architecture Comparison.** Here, effectiveness of four different *RBFN* and five *MLP* architectures was compared. Training dataset included the same working days of the week from three consecutive weeks (e.g. March, 10, 17 and 24). The test data was from the following week (March 29 to April 2). Only data from working days was used. The RBFN architectures were implemented using the RBF [14] code. The hidden layer used Gaussian radial basis function:  $exp(-\beta r^2)$ . The tested architectures had  $M = 10, 15, 25, 35$  neurons in the hidden layer (denoted as RBFN  $M$ ). As can be seen in [15], for short routes (*Center–Praga*, *short within Center*, *short within periphery*) and *Express* routes, the most accurate results were obtained by “smaller” RBFN architectures (RBFN 10 and

RBFN 15). For the remaining routes, RBFN 25 and RBFN 35 performed better. Overall, if one model is to be selected, RBFN 25 seems to be the “best architecture”.

For MLP, five architectures have been tried: networks with 2, or 3, hidden layers and: [6, 12], [12, 32], [6, 8, 12], [8, 8] neurons. Here, *ReLU* and *tanh* were also compared, as activation functions. The results are presented in Table 2.

Overall, the most effective, and stable, architecture, for all groups, had two hidden layers (12 and 32 neurons), with ReLU as the activation function. Since, the [12, 32] ReLU was the “overall winner”, its performance is reported in what follows. Additional results, comparing performance of RBFN 25 and MLP [12, 32] ReLU, can be found in [15].

**Table 2.** MLP-based prediction results

Grupa	Model									
	MLP 6, 12 ReLU		MLP 12, 32 ReLU		MLP 12, 32 ReLU tanh		MLP 6, 8, 12 ReLU		MLP 8, 8 tanh	
	MAE	STD	MAE	STD	MAE	STD	MAE	STD	MAE	STD
Center-Praga	112.33 s	143.9 s	96.43 s	119.94 s	<b>94.08 s</b>	<b>118.30 s</b>	114.5 s	147.78 s	115.93 s	146.75 s
Center-periphery	<b>101.54 s</b>	<b>133.53 s</b>	106.9 s	135.26 s	110.86 s	146.74 s	110.69 s	145.8 s	143.31 s	180.73 s
Long within periphery North-South	139.06 s	178.95 s	119.79 s	<b>159.51 s</b>	<b>114.39 s</b>	164.76 s	120.05 s	164.01 s	136.25 s	173.61 s
Long within periphery East-West	122.47 s	167.79 s	<b>113.67 s</b>	<b>151.78 s</b>	133.6 s	181.72 s	124.23 s	162.63 s	154.59 s	196.67 s
Long through the centre	109.35 s	146.14 s	<b>97.71 s</b>	<b>128.49 s</b>	118.17 s	153.16 s	105.89 s	139.59 s	109.16 s	143.21 s
Express	127.52 s	168.45 s	<b>92.46 s</b>	<b>123.74 s</b>	118.71 s	157.53 s	116.61 s	154.02 s	132.97 s	174.49 s
Short within center	97.02 s	130.73 s	<b>95.42 s</b>	<b>121.13 s</b>	108.07 s	139.06 s	106.31 s	132.19 s	101.55 s	135.38 s
Short within periphery	107.13 s	133.08 s	<b>90.24 s</b>	<b>122.65 s</b>	96.44 s	124.07 s	99.89 s	129.8 s	104.84 s	139.98 s
All groups	114.55 s	150.32 s	<b>101.57 s</b>	<b>132.81 s</b>	111.78 s	148.16 s	112.27 s	146.97 s	124.82 s	161.35 s

Next experiment used HA method [1, 5], which applies average travel times from previous days to estimate the current travel time. For each bus line, data from the training sets (i.e. all working days from March, 8–26) was divided into 20-min groups (i.e. records from 10:00–10:20 belonged to one group). The average travel times between current locations and all stops, till the end of the route, were calculated. Average travel times, determined using this algorithm, have been stored as the training sets. Next, for each bus line, for all test sets (i.e. data for working days from March, 29 to April, 2), travel time predictions were calculated using the HA algorithm. In addition, analogous calculations were carried out, while the average travel times were calculated for the same day of

**Table 3.** Prediction MAE for timetable, HA for same days, HA for working days.

Group	Time table	HA(the same day)	HA(all days)
Long within periphery North-South	79.22 s	81.53 s	<b>71.94 s</b>
Long within periphery West-East	80.85 s	75.02 s	<b>62.38 s</b>
Center-periphery	64.76 s	62.02 s	<b>61.62 s</b>
Long through the centre	71.58 s	68.75 s	<b>67.64 s</b>
Express	60.38 s	64.65 s	<b>52.70 s</b>
Center-Praga	64.56 s	52.89 s	<b>50.22 s</b>
Short within periphery	55.87 s	50.92 s	<b>35.65 s</b>
Short within centre	60.41 s	53.02 s	<b>49.02 s</b>

the week only. For example, travel time predictions for March 31 (Wednesday) were based on data from March 10, 17, 24. Table 3 summarizes the MAE values of travel time predictions using the HA method. As can be seen, HA based on data from all working days provides better accuracy than when using data from the same day of the week only.

Next, a hybrid model was developed, consisting of: (1) RBFN 25 or MLP [12, 32], using predicted travel time based on schedule data and delay at the last stop; and (2) RBFN 25 or MLP [12, 32], using estimated travel time using HA. When making predictions, depending to which group given bus line belonged, and the distance for which the prediction was performed, the hybrid approach used model that was expected to be the best for that combination (bus+distance).

The created hybrid model was compared with: MLP and RBFN models that used the basic set of features, the HA algorithm, and predictions based on data from timetables only. Figure 1 represents comparison of MAE values (in seconds) of the predictions made by the hybrid model with the remaining methods, depending on the distance (number of stops). Due to space limitation, only two, very different, cases are reported. Bus lines belonging to the same group had different lengths of routes. Therefore, black triangles mark distances for which the prediction of travel time ends for specific bus lines, belonging to the group described by the graph. For this reason, the graphs report rapid changes in MAE for adjacent distances, as in the case of *Long within periphery* group, where lines have total of 30, 35, 36 and 38 stops. Results represented in both figures, and the remaining experimental results (see, [15]), show that for short-distance predictions, for all groups, the HA algorithm combined with distribution times, delivered better accuracy than the hybrid model.

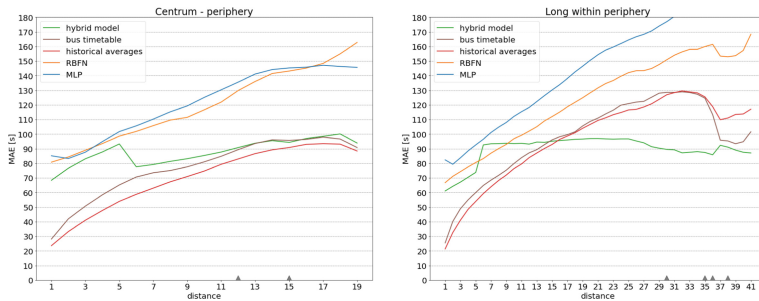


Fig. 1. Comparison of the MAE value of prediction methods at different distances

### 4 Concluding Remarks

Now, let us consider top 5 lowest MAE values found in the literature vis-a-vis results reported here: (1) ANN from [2]; MAE at 40s; for distances of up to 16 stops. (2) Hybrid model reported here; MAE at 67.17s; for the *Short through the center* group, for time interval 19:00 to 23:00. (3) ANN from [1]; MAE at

70 s. (4) BP from [11]; MAE at 125 s; for morning rush hours. (5) MLP from [5]; MAE at 138.40 s; for travel time of trams for noon hours.

In this context recall that in each article, results were obtained for bus (or tram) lines from different cities, with very different road and traffic structures, and public transport characteristics. For example, in more populated cities, or those with less developed infrastructure, there may be more delays due to the heavy traffic, which may influence the accuracy. Moreover, Trondheim (city population is around 200 thousands, while metropolitan population is around 280,000) is much smaller than other cities. Further, Warsaw is split by a river, with limited number of bridges. Besides, analyzed bus/tram lines had different lengths. Finally, no other work used data for multiple bus lines “jointly” (combined into groups). Taking this into account, it can be argued that the results reported in this contribution (and in [15]) are very competitive and worthy further explorations.

Separately note that additional experiments confirmed that use of auxiliary features, e.g. number of busses moving between stops, or number of crossings with lights, did not visibly improve accuracy of prediction for any of the tried models. This fact seems to be somewhat counter intuitive, but it supports results reported in [2].

## References

1. Fan, W., Gurmu, Z.: Dynamic travel time prediction models for buses using only GPS data. *Int. J. Transp. Sci. Technol.* **4**(4), 353–366 (2015). <https://www.sciencedirect.com/science/article/pii/S204604301630168X>
2. Dahl, E., Sjøfjell, A., Skogen, S.: On implementations of bus travel time prediction utilizing methods in artificial intelligence, NUST (2014)
3. Sjøfjell, A., Dahl, E., Skogen, S.: Intelligent transportation systems and artificial intelligence - a state of the art review, NUST (2013)
4. Lin, Y., Yang, X., Zou, N., Jia, L.: Real-time bus arrival time prediction: case study for Jinan, China. *J. Transp. Eng.* **139**(11), 1133–1140 (2013)
5. Zychowski, A., Junosza-Szaniawski, K., Kosicki, A.: Travel time prediction for trams in Warsaw. In: Kurzynski, M., Wozniak, M., Burduk, R. (eds.) *CORES 2017*. AISC, vol. 578, pp. 53–62. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-59162-9\\_6](https://doi.org/10.1007/978-3-319-59162-9_6)
6. Yang, J.S.: Travel time prediction using the GPS test vehicle and Kalman filtering techniques. In: *Proceedings of the 2005, American Control Conference, 2005*, pp. 2128–2133. IEEE (2005)
7. Jiwon, M., Kim, D., Kho, S., Park, C.: Travel time prediction using k nearest neighbor method with combined data from vehicle detector system and automatic toll collection system. *Transp. Res. Rec. J. Transp. Res. Board* **2256**, 51–59 (2012)
8. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA (1998)
9. Shaji, H.E., Tangirala, A.K., Vanajakshi, L.: Prediction of trends in bus travel time using spatial patterns. *Transp. Res. Procedia* **48**, 998–1007 (2020)
10. Comi, A., Zhuk, M., Kovalyshyn, V., Hilevych, V.: Investigating bus travel time and predictive models: a time series-based approach. *Transp. Res. Procedia* **45**, 692–699 (2020)

11. Fei, J., Lu, Y., Guo, Y., Zhang, H.: Predicting bus arrival time using BP neural network and dynamic transfer. *Procedia Comput. Sci.* **174**, 95–100 (2019)
12. Wang, L., Zuo, Z., Fu, J.: Bus arrival time prediction using RBF neural networks adjusted by online data. *Procedia. Soc. Behav. Sci.* **138**, 67–75 (2014)
13. Wang, L., Zuo, Z., Fu, J.: Bus dynamic travel time prediction: using a deep feature extraction framework based on RNN and DNN. *Electronics* **9**(11), 1876 (2020). <https://doi.org/10.3390/electronics9111876>. <https://www.mdpi.com/2079-9292/9/11/1876>
14. Vidnerová, P.: RBF-Keras: an RBF Layer for Keras Library (2019). [https://github.com/PetraVidnerova/rbf\\_keras](https://github.com/PetraVidnerova/rbf_keras)
15. Pałys, L., Ganzha, M., Paprzycki, M.: Applying machine learning to predict behavior of bus transport in Warsaw, Poland. <https://arxiv.org/submit/4254888>