Machine learning for bus travel prediction

Łukasz Pałys[1], Maria Ganzha[1][0000−0001−7714−4844], and Marcin Paprzycki[2][0000−0002−8069−2152]

[1] Warsaw University of Technology, Warsaw, Poland
maria.ganzha@pw.edu.pl
[2] Systems Research Institute Polish Academy of Sciences
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 problems has been discussed in the literature. Applied methods can be divided into groups. First, *statistical methods*, such as k-NN [8, 2], regression [2], prediction based on sequence patterns [10] or time series [11], 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: back propagation neural network [12], radial basis function network [13], multilayer perceptron [5, 1]. The last group consists of *hybrid methods* that combine multiple algorithms into a single model [4, 14]. Let us now present selected results, while [16] contains detailed discussion.

The Pattern Sequence-based Forecasting was used in [10], 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 “satisfatory”.

Time series approach was used in [11] to forecast bus travel time in Lviv (Ukraine). The average error, for travel time towards city center was approximately 3 minutes, and 2 minutes in the opposite direction.

In [13], 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 [14], 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

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

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 seconds. 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 prediction**s 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*[3], exact location of public buses, reported in real time, is available. From there, 30 days of data about bus move-

[3] https://api.um.warszawa.pl/

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

ments 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[4]. 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) 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 experimental results[5]. 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 previous

[4] https://www.wtp.waw.pl

[5] All reported models have been implemented in Python, using Keras library

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

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

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

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

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.

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 [15] 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 [16], 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

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

Table 2. MLP-based prediction results

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

follows. Additional results, comparing performance of RBFN 25 and MLP [12,32] ReLU, can be found in [16].

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-minute 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 the week only. For example, travel time predictions for March 31 (Wednesday) were based on data 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

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	79.22s	81.53s	**71.94s**
Long within periphery West-East	80.85s	75.02s	**62.38s**
Center–periphery	64.76s	62.02s	**61.62s**
Long through the centre	71.58s	68.75s	**67.64s**
Express	60.38s	64.65s	**52.70s**
Center–Praga	64.56s	52.89s	**50.22s**
Short within periphery	55.87s	50.92s	**35.65s**
Short within centre	60.41s	53.02s	**49.02s**

on data from all working days provides better accuracy then 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).

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

The created hybrid model was compared with: MLP and RBFN models that used 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 remain-

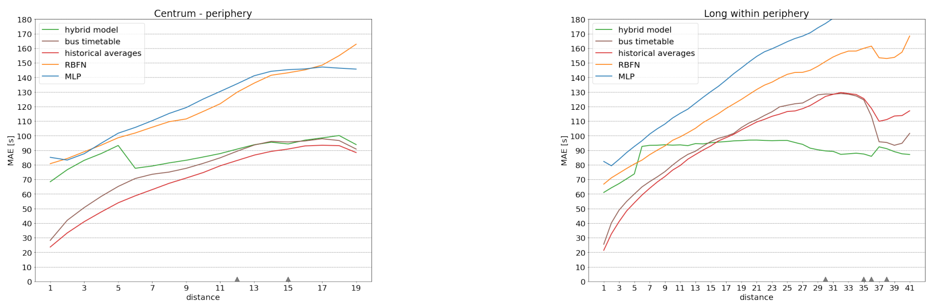

Fig. 1. Comparison of the MAE value of prediction methods at different distances

ing experimental results (see, [16]), show that for short-distance predictions, for all groups, the HA algorithm combined with distribution times, delivered better accuracy than the hybrid model.

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 70s. (4) BP from [12]; MAE at 125s; for morning rush hours. (5) MLP from [5]; MAE at 138.40s; 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,

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72

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 [16]) 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. W. Fan, Z. Gurmu, *Dynamic Travel Time Prediction Models for Buses Using Only GPS Data*, Int. J. of Transportation Science and Technology, 2015, 353-366.
2. E. Dahl, A. Sjåfjell, S. Skogen, *On Implementations of Bus Travel Time Prediction Utilizing Methods in Artificial Intelligence*, NUST, 2014
3. A. Sjåfjell, E. Dahl, S. Skogen, *Intelligent Transportation Systems and Artificial Intelligence – a State of the Art Review*, NUST, 2013
4. Y. Lin, X. Yang, . Zou, L. Jia, *Real-Time Bus Arrival Time Prediction: Case Study for Jinan, China*, Journal of Transportation Engineering, 2013
5. A. Żychowski, K. Junosza-Szaniawski, A. Kosicki, *Travel time prediction for trams in Warsaw*, CORES, 2017
6. Yang, J.S., *Travel time prediction using the GPS test vehicle and kalman filtering techniques.* In: Proc. of the Am. Control Conf., 2005, IEEE(2005) 2128–2133
7. C. Wu, J. Ho, D. T. Lee, *Travel-Time Prediction With Support Vector Regression*, IEEE Transactions on Intelligent Transportation Systems 5(4): 276-281, 2005
8. M. Jiwon, D. Kim, S. Kho, C. Park, *Travel Time Prediction Using k Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System*, Transportation Research Record Journal of the Transportation Research Board 2256:51-59, 2012
9. Haykin, S., *Neural Networks: A Comprehensive Foundation. Prentice Hall PTR*, Upper Saddle River, NJ, USA (1998)
10. H. E. Shaji, A. K. Tangirala, L. Vanajakshi, *Prediction Of Trends In Bus Travel Time Using Spatial Patterns*, Transportation Research Procedia 48 998–1007, 2020
11. A. Comi, M. Zhuk, V. Kovalyshyn, V. Hilevych, *Investigating bus travel time and predictive models: a time series-based approach*, Transportation Research Procedia 45 692–699, 2020
12. J. Fei, Y. Lu, Y. Guo, H. Zhang, *Predicting Bus Arrival Time Using BP Neural Network and Dynamic Transfer*, Procedia Computer Science 174 95-100, 2019
13. L. Wang, Z. Zuo, J. Fu, *Bus Arrival Time Prediction Using RBF Neural Networks Adjusted by Online Data*, Procedia – Soc. and Beh. Sc. 138, 67-75, 2014
14. L. Wang, Z. Zuo, J. Fu, *Bus Dynamic Travel Time Prediction: Using a Deep Feature Extraction Framework Based on RNN and DNN*, Electronics, 2020
15. P. Vidnerová, *RBF-Keras: an RBF Layer for Keras Library.*, 2019, Available at https://github.com/PetraVidnerova/rbf_keras
16. Ł. Pałys, M.Ganzha, M. Paprzycki, *Applying machine learning to predict behavior of bus transport in Warsaw, Poland*, at https://arxiv.org/submit/4254888

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08754-7_72