BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2026
http://excel.fit.vutbr.cz

Prediction and visualization of public transport delays

Martin Kováčik*

Abstract
Public transportation is an indispensable part of modern cities, and great emphasis is placed on its punctuality. The goal of this project is to create a delay prediction tool that will help passengers choose their connections. Although not all types of delays—such as traffic accidents, breakdowns, etc.—can be predicted, there are predictable factors that regularly influence delays and create certain patterns in vehicle behavior. The result of this work is a predictor that identifies patterns in these factors and returns the most accurate results possible. The implemented model achieves decent results, with a mean absolute error of ±40 seconds. This tool will be appreciated not only by passengers who use it to predict their specific connections, but also by transportation company employees for various analyses, planning, and experimentation with the system. For this and other purposes, the predictor generates a route for the requested connection in a format suitable for visualization.

*xkovacm01@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

A large portion of the urban population chooses public transportation as their primary mode of travel. They use it to get to school and work, and sometimes for important appointments such as job interviews, where being late can be a **major problem** and may **cost them a job opportunity**.

Gathering all the necessary factors and data for training is no easy task. A practical solution must therefore account for **various negative factors found in real-world data**, such as anomalies, missing or incomplete records, and many other challenges. The advantage of this tool is that it is not only appreciated by passengers themselves, but can also be used by other analysts for decision-making, thereby providing them with a broader perspective on the system as a whole.

Most commercial solutions today focus specially on **real-time predictions**, typically based on the vehicle's current delay, which they **simply add to** the estimated arrival or departure time, or they simply indicate whether a delay is expected based on historical data. One of the largest systems in our region is the IDS JMK[1] interactive map, which displays the current locations of vehicles. Each vehicle is listed with its current delay, which is displayed at all stops **without taking into account various factors** such as weather, rush hour, and others. Then there's IDOS[2], which, through its search engine, can display a historical delays at a selected stop. This solution is a bit better, but it's more **challenging for passengers**, who have to estimate the potential delay on their own.

Provided tool is designed so that its model has been trained on **historical data** and **other factors** to best represent the current state of the system for prediction purposes. Furthermore, it uses **real-time weather data** from the weather station closest to the requested stop, via the OpenWeatherAPI[3]. The tool always predicts the entire route of the prediction line, as delays accumulate.

The implemented solution allows passengers to **predict delays not only at a specific stop**, but also along **the entire route**. It also provides insight into how the system may behave on different days, thereby offering additional guidance for decision-making when planning routes and other related decisions. The main challenge of this solution is the **availability of real-time GTFS**[4] transport data and the shortcomings of this standard, which must be filtered out and corrected before the model is trained.

[1] https://mapa.idsjmk.cz

[2] https://idos.cz

[3] https://openweathermap.org/api

[4] https://gtfs.org/documentation/realtime/reference/

2. Design of the Solution

The tool consists primarily of **three parts**. The first part involves **downloading and encoding** data from Lissy[5], a software program that stores historical traffic data. The downloaded data is **parsed, stored and encoded** as shown in Figure 1. The next part involves **training the neural network** itself, as described in section 2.2. Finally, there is the third part, which is where the **prediction of the resulting delay** actually takes place, whether for a single stop or the entire route, more details are provided in 2.3.

2.1 Data Download and Encoding

This section primarily focuses on **retrieving traffic and weather data** for the specified time period. Data is always downloaded on a per-line basis, where the closest of the five available weather stations is selected for the requested stop, and an API query is sent for that specific time. The response is then **stripped of unnecessary data**, and the result is a parsed object, as shown in Figure 1.

Next, information about the route is retrieved—both **static data** (such as the connection details, route, and vehicle type) and **dynamic data** for the specified time, in this case the delay. The data is then enriched with context and **appropriately normalized** using the scikit-learn library[6].

2.2 Neural network training

The training process itself depends on the **quality of the available open data**. The neural network itself contains two hidden layers, the **SiLU function** is used as the nonlinear activation function, as it is better suited for predicting delays. Since the data-extraction component already normalizes and **stores the data in normalized form**, this step is omitted from the training process. The model was trained on data covering the last two months.

The dataset is split so that **80% is used for training and 20% for testing**, the model tracks both training and testing losses (MSELoss, MAELoss). Training is set to a maximum of 100 epochs, but the training itself includes the **ReduceLROnPlateau scheduler** to reduce the learning rate. The patience parameter is set to 3, and if there is no improvement in MAE on the test data after 7 epochs, **the best model so far is saved**.

2.3 Prediction

The prediction itself is performed in **two ways**. These two methods differ primarily in whether the user receives the entire route and its shape id, which is compatible with the aforementioned **Lissy software**, or only the predicted value for the requested stop.

It is important to note that in both cases, the **prediction is made for the entire route**, because **delays accumulate** in most cases, therefore, one of the inputs to the neural network is the previous delay, and the first predicted segment assumes that this delay is 0 (assuming that the train departs without delay). Therefore, in addition to the first prediction, **each subsequent predicted value serves as the input for the next section**.

Visualization

If the user wants to retrieve the **entire route**, they set the visualization parameter to true in the input and **do not need to specify which stop** they want to predict. This input is shown in Figure 2. The predictor **retrieves the necessary data** to construct the required input for the neural network and returns the shape ID and the entire predicted route to the user. This mechanism therefore assumes that the delay with which the vehicle arrived at the stop **best represents the entire segment**.

Individual delay

Another option is to generate a prediction for a single stop. If the user wants this result, they set the visualization to false and add the desired stop, this input is shown in Figure 2. As in the previous case, the predictor retrieves the necessary context and predicts the entire route, but returns only the delay at the desired stop.

3. Results

The result of this work is a predictor with an MAE of **approximately ±40 seconds** on the test data. It operates in two modes: **visualization mode and individual prediction mode**. The visualization format was adapted to the aforementioned Lissy software so that this visualization could be interpreted as a **geographic map**. An overview of this visualization is shown in Figure 3, with examples of two predicted routes.

Individual predictions can then also be used by other route planners to provide passengers with additional information about the connection they are looking for and **help them make a decision**.

Acknowledgements

My sincere thanks also go to my advisor, Ing. Juraj Lazúr, for his time and expert assistance in providing and helping me navigate the transportation data.

[5] https://dexter.fit.vutbr.cz/lissy
[6] https://scikit-learn.org/stable/