Predicting train arrival delays - evaluation of different input features

GEO 511 Master's Thesis

Author
Olivier Niklaus
11-945-532

Supervised by
Dr. Haosheng Huang

Faculty representative
Prof. Dr. Robert Weibel

01.10.2019
Department of Geography, University of Zurich

MASTER THESIS

Predicting train arrival delays – evaluation of different input features

30. SEPTEMBER 2019
OLIVIER NIKLAUS
11-945-532

Supervised by
Dr. Haosheng Huang
Geographisches Institut Zürich

Faculty member
Prof. Dr. Robert Weibel
Geographisches Institut Zürich

Universität Zürich UZH

I would like to thank everybody, who supported me during the process of creation of this thesis. Special thanks go to my family and especially to my beloved for always being with me......

Abstract

In the era of digitalization and location-based-services, public transportation operators face new challenges in order to meet the demands of public transportation users. On the one hand this includes on demand transportation services as car- and bicycle-sharing. On the other hand, public transportation users demand reliable transportation-services in order to plan their activities in busy calendars. Furthermore, reliable transportation services are also important for the economic success of different regions. Consequently, from this side urban planners and decision-makers claim new demands to public transportation operators. Big data in public transportation has opened new possibilities for public transportation operators to meet these demands with innovative applications. Such applications do not necessarily need to directly target the end-customer. Rather big data applications can also help transportation operators improve their services, which in turn satisfies the end-customer. One such application would be the ability to effectively predict delays of transportation services. This would allow transportation operators to make the right decisions in case of delays in order to maintain successful transportation operations. Consequently, this has led to the development of different delay prediction systems using different approaches and techniques.

The aim of this thesis is to contribute to a better understanding how big data can contribute to predict train arrival delays. In current research different input variables and machine-learning techniques have been evaluated to predict arrival delays in public bus transportation and railway transportation networks. The variables used for delay prediction have been characterizing specific trip-related properties of the transportation service. Only recently, the focus has shifted from trip-related properties towards variables derived by a more holistic network-approach (Oneto *et al.*, 2018; Sun *et al.*, 2018). Using this approach different useful variables for transportation delay prediction that are in relation with the whole transportation network have been identified. However, until now little research has been conducted on how these variables contribute to arrival delay prediction. Within this thesis it could be found that not all input variables discussed in existing literature contribute to delay prediction in the same way. Results show that timetable related input variables do only have a marginal contribution to delay prediction. Whereas variables capturing the current traffic situation within the network highly contribute to successful prediction of arrival delays in railway transportation systems. The aim of this thesis is to gain a deeper understanding in the process of feature creation. More specific, the goal is to identify how, and which kind of features contribute more or less to train arrival delay prediction.

Table of Contents

1. Introduction

1.1 Motivation and background

Reports in the late 1960s a majority of Swiss citizens considered public transport as a discontinued model, which should be maintained until the full motorization of Swiss citizens and for poor and elderly people. The report presented in 1972 identified that an inadequate level of service, especially consisting of inconvenient timetables and especially unpunctuality, as the railways' main weaknesses. As a result, the authors proposed the implementation of a nation-wide, regular interval 'pulse' timetable called "Taktfahrplan" (Meiner, 1991; Petersen, 2016). Nowadays, public transportation is an essential part of communities and cities and whole nations. Switzerland is known for having among the highest rates of public transport use in Western Europe, as well as nationally-coordinated scheduling that extends deep into rural areas (Petersen, 2016). Furthermore, timetabling is still considered to take in a key-role for effective public transportation (Lee, Yen and Chou, 2016). Today 17% of all commuters in Switzerland use the public train system to reach their workplace. 57% of this 655'000 people use public trains to travel for more than 50km to their workplace (Bundesamt für Statistik, 2016). In reality the number of persons using the public train transport for long distances must be considered even higher, as the famous "Pendlerstatistik" provided by the Bundesamt für Statistik (BFS) in 2016 does not include train passengers using public transport for other purposes, for example leisure-activities (Bundesamt für Statistik, 2016). The mentioned 50km corresponds approximately to half the distance from the popular train-route Zürich-Bern. The distance covered by this train-route might not be of high importance in other countries but for sure it is within Switzerland. One need to consider that within approximately 100km the train network connects three political units of state-level (Cantons) and connects the economically most important region of Switzerland (Zürich) with its national capital Bern. This highlights the political, economical and social importance of the long-distance railway traffic network in Switzerland. In order to minimalize negative economical and social effects it is important that the transportation service is continuously operated with as little disruptions as possible.

The backbone of the Swiss national public transport system is the "Fernverkehrs-Netz", which consists of the long-distance railway traffic. The long-distance traffic concession is granted by the Swiss government to the Schweizerischen Bundesbahnen (SBB). The most central aim for the long-distance traffic is to connect all areas of action and superordinate centers of Switzerland and integrating Switzerland into the European major traffic axis (Bundesamt für Verkehr BAV, 2017). The long-distance railway network serves all regions within Switzerland and provides the reference pulse signal for the nation-wide pulse timetable (Schweizerische Bundesbahnen, 2017). Therefore, service disruptions of the long-distance traffic affect the whole public transportation network in Switzerland. This emphasizes the importance that the long-distance traffic is on schedule in order to minimalize negative effects on economy and society.

1.2 Problem and goal setting

As outlined above it is important that public trains operated by the SBB are on schedule to guarantee transit connections with other more regional operating public transport agencies. In a highly scheduled railway traffic networks a single delayed train may cause a domino effect of secondary delays over the entire network, which is a main concern to railway planners and operations dispatchers (Goverde, 2010).

Therefore, the development of a system predicting train delays in the swiss long-distance railway network would help to improve the public transport system as a whole and its supervisor in decision-making. Additionally, it could be a mean to provide customers with more detailed real-time information, which would increase the customer's perception of reliability of the railway transportation service. Moreover, in case of service disruptions it could provide valid alternatives to passengers looking for the best train connections (Dotoli *et al.*, 2017). The goal of this master thesis is to contribute to the current state of research in assessing and enhancing public transportation network services.

2. Related literature

2.1 Big data analytics and public transportation

The fast-paced development of advanced technologies has led to the accumulation of a vast amount of gathered data. This development did not stop either in the domain of public transportation. The properties of big data can be characterized by the 4 V's. Namely, volume, variety, velocity and value (Fosso Wamba *et al.*, 2015; Ghofrani *et al.*, 2018; Neilson *et al.*, 2019). All of them describe specific properties of the data and its related challenges when facing big data. 'Volume' is the term for describing the magnitude of data available in big data analytical tasks. A major challenge with large amounts of data lies in its computational processing and storage of the data. Consequently, new computational frameworks, such as scalable distributed computing have been developed to cope with this challenge (Neilson *et al.*, 2019). 'Variety' refers to the various sources from which data can be generated. The variety of data sources can range from sensors, which measure temperature over to mobile devices registering spatial movement over time or even social media posts. Mostly these data sources have developed a data structure that is senseful within their source-system, therefore a major challenge regarding big data variety, lies in integrating different data structures in a meaningful way, where the data from different sources can be analyzed and processed (Neilson *et al.*, 2019). The speed or frequency of generating data is characterized by 'Velocity'. The challenge here is to find ways to treat data corresponding to their purpose. As Assunção et al. (2015) highlights, data can arrive and require processing at different speeds. While for some analytics applications, the arrival and processing of data can be performed in batch, other applications require continuous and real-time analyses.

2.1.1 Big data analytics and potential application fields in transportation

In the domain of public transportation big data has the potential to improve the safety and sustainability of transportation systems and offers opportunities to apply evidence-based approaches to decision-making. Neilson et al. (2019) emphasize three main ideas, how institutions responsible for transportation can make use of big data to create safer and more sustainable transportation systems.

First, an obvious way to improve the sustainability of transportation systems is to share real-time information with users. This enables users to make information-based decision to solve routing problems. In addition providing public transportation operators with real-time information might be able to react faster to service disruptions and allow improved decision making (Ghofrani *et al.*, 2018).

Secondly, Neilson et al. (2019) highlights that analysis of past transportation data can support evidence-based approaches for urban planning and enhance decision makers' understanding of a public transportation network. For example, by analyzing data from the automatic smart card fare collection system in Singapore, which contains origin-destination pairs, Zhong et al. (2014) could comprehend how people use the local metro system and therefore understand how people move within the city's metro system. Moreover, by analyzing the data over several years they were able to detect changing patterns in peoples' movement over space and concluded that the reason for this change is likely to be because of the extension of the metro system in specific areas. This example showcases how big data analytics can contribute to city planners and transportation network planners understanding in people's behaviour and therefore support their decisions about future investments.

The third main idea mentioned by Neilson et al. (2019) is to improve safety in transportation systems by analyzing collisions or near misses. This idea especially refers to motorized private transport, where location and types of traffic collisions can be registered, aggregated and analyzed to identify high-collision areas or the spatial distribution of certain collision types (Xie and Yan, 2013; Shafabakhsh, Famili and Bahadori, 2017). Nevertheless, this idea is also a topic in public transportation systems, even within rail transportation, which is currently the safest mode of surface transportation. However, accidents still occur and analyzing historical accident data can provide useful, high-level views regarding safety trends and characteristics. But as accident datasets of railway networks are usually small it is hard to depict and predict the localized risk profile for a specific location given a time period (Ghofrani *et al.*, 2018).

Ghofrani et al. (2018) highlighted that current research of big data analytics in railway systems context can be categorized into three main application fields. The first category focuses on maintenance. Big data analytics in railway maintenance focuses on how big data analytics can contribute the activity of maintaining the functionality of system components such as vehicles, signaling equipment and tracks (Li *et al.*, 2014; Fumeo, Oneto and Anguita, 2015). The second application field Ghofrani et al. (2018) denotes is the field of operations. They emphasize that intelligent rail transportation systems that contain big data analytics have provided innovative technologies for railway infrastructure managers and train operation companies that help them to make more efficient decisions (Liang, Martin and Cui, 2017). In this field big data analytics supports decision-makers in real-time for rail-traffic management by improving timetabling and the creation of simulation models. This application field is much related to Neilsons et al's (2019) first and second idea on how big data can improve transportation systems. According to Ghofrani et al. (2018), the third main application field of big data analytics in railway systems is dedicated to safety. As already mentioned, rail transportation is currently the safest transportation mode, but to achieve this railway infrastructure and operations needs to be monitored, which involves big data sources and consequently a deep understanding of the data.

Application fields of big data analytics in bus transportation domain are similar to those in railway systems. Similarly to railway systems the field of operations in bus transportation system has benefited from big data analytics (Moreira-Matias *et al.*, 2015). Bus transportation systems have especially benefited since the integration of automatic vehicle location and automatic passenger counting systems into the dispatching systems. The availability of this data opened new research directions for improving the bus transportation services' reliability (Sun *et al.*, 2018).

The focus of this thesis is to evaluate a new approach of predictive analytics in the operations-field of public transportation. In current research predictive analytics as a sub-field of big data analytics is the most dominant approach in operations-related research (Ghofrani *et al.*, 2018). Predictive analytics provides tools and methods to make predictions about future events by analyzing current and historical data and therefore is a promising approach to encounter scheduling problems in transportation systems (Assunção *et al.*, 2015; Ghofrani *et al.*, 2018).

2.1.2 Predictive analytics and public transportation scheduling

Törnquist (2006) highlights that railway traffic scheduling is often considered a difficult problem primarily due to its complexity regarding size and the signification interdependencies within railway network. In public bus transportation systems improper scheduling is an important internal factor causing reliability problems of the bus transportation service (Moreira-Matias *et al.*, 2015). Törnquist (2006) reviewed 48 different approaches to approach scheduling problems. For this she made a distinction between scheduling (timetabling) and re-scheduling (dispatching). Scheduling is the act of constructing a scheduling from scratch, while re-scheduling indicates that a schedule already exists and will be modified according to deviations from the present schedule. Furthermore, she states that scheduling can be carried out with different time perspectives. Tactical scheduling refers to creating master schedules that specify a strict route and timetable for each train with the intention to execute it in real-time. It usually involves scheduling for large traffic network for a long-time horizon and may be more complex, as on the one hand it needs to reflect the demand of several stakeholders on the one side. On the other hand, tactical scheduling must also consider infrastructural limitations of the railway network. Operational scheduling on the other hand has a short time frame and is initialized close to the time of the public transportation service's departure.

Applying techniques and methods of predictive analytics could contribute to a better understanding of public transportation networks and their inherent dynamics. More specifically, predictive analytics offer new possibilities to solve schedule planning and dispatching problems in public transportation. Predictive analytics provides methods and to techniques to build models, which predict public transportation delays or travel time in short-term or even real-time. This can offer possibilities to detect potential instabilities in-time and alert dispatchers to reschedule specific trains or take other actions, if deviations from the schedule occur. This would help to maintain the reliability of the system on the on hand but also customer satisfaction on the other. Therefore, predictive analytics can support operational scheduling by verifying different scenarios and provide the foundation for evidence-based decision-making. That in turn is congruent to Neilson et al.'s (2019) first idea how big data analytics can be applied in public transportation. In railway networks, such models use train movement data that is collected from infrastructure track occupation records, sensors or mobile GPS devices gathered in real-time. Using this incoming stream of data, the model then tries to predict whether an arrival is delayed or not (Ghofrani *et al.*, 2018). Similar approaches have been undertaken in public bus networks (Oruganti *et al.*, 2016; Gal *et al.*, 2017; Sun *et al.*, 2018). The domain of public bus network has especially profited since transit agencies have been integrating real-time sensors into public transportation systems.

Predictive analytics often make use of machine learning algorithms (Dey, 2016). In general, these algorithms are trained to predict a specific value. The algorithm gets trained by providing it a large amount of data. The data includes different variables that may contribute to the value the algorithm needs to predict. During the training phase the algorithm tries to find the underlying function to predict the variable of interest. Currently a large variety of learning algorithms have been assessed to predict delay. Most popular are artificial neural networks (ANN), Support Vector Regression (SVR), Random Forest Models and Gradient Boosting Decision Trees (GBDT) (Milinković *et al.*, 2013; Moreira-Matias *et al.*, 2015; Oruganti *et al.*, 2016; Ghofrani *et al.*, 2018; Yamaguchi, As and Mine, 2019). As suggested by Neilson et al. (2019) one can make use of historical transportation data containing service delay information to train learning algorithms, which in turn can be used for schedule evaluation on a tactical or operational scheduling tasks. The used algorithms and data are often similar for both scheduling tasks and therefore a distinct differentiation between predictive analytics on an operational or tactical level is not always possible. Moreover, the same models even can be applied on both levels (Peters *et al.*, 2005).

As Törnquist (2006) highlights, tactical scheduling usually involves scheduling for a large traffic network for a long time horizon and the time available for creating the timetable may be several months. Therefore, using predictive analytics on a tactical level should approach the transportation network in a more holistic way. According to Marković et al. (2015) the aim of applying predictive analytics on a tactical level is to determine the functional relation between transportation system characteristics and a variable of interest, for example delayed arrivals or travel-time (Marković *et al.*, 2015). In order to account the complexity regarding size and interdependencies between transportation vehicles, predictive analytics on a tactical level requires a more holistic perspective on the transportation network. Furthermore, if predictive analytics are applied in the context of transportation delays it is worth noting the intrinsic time varying nature of the delay phenomenon (Oneto *et al.*, 2017). The resulting models from applying predictive analytics on a tactical level can then be used for tactical planning such as timetabling and resource planning and the evaluation of different scenarios (Marković *et al.*, 2015).

The following section presents different applications of predictive analytics. Furthermore, current state-of-the-art techniques and approaches are presented. The aim is to provide an overview of current research related to different delay and travel time prediction approaches for bus and train transportation networks.

2.2 Delay prediction in bus transportation

2.2.1 Available data sources and general problem setting

Public transportation in urban areas is a critical component of a smart and connected community (Sun *et al.*, 2018). The advantage of bus transportation systems is that they can be integrated within a road network. Considering an existing road network, buses need less infrastructural investments, compared to tram or train transportation system. However, the downside of this, is that buses share their network with other participants of the road network. This can lead to network overloads and restrict the reliability of the bus transportation system as services are cannot be on time. Consequently, monitoring their services is getting more relevant for bus transportation agencies. Recently, bus agencies have been integrating real-time sensors into their dispatching systems (Sun *et al.*, 2018). Most often these systems rely on the use of the Global Positioning System (GPS) for capturing a vehicles position. GPS is traditionally the basis of automatic vehicle location (AVL), which is one component of a traditional bus agency dispatching system. This component automatically registers a vehicles speed and location in latitude-longitude pairs within an interval of 10-30s and broadcasts it (Moreira-Matias *et al.*, 2015). Now, all the incoming data can be stored or directly used and processed to provide arrival time predictions at bus stops (Sun *et al.*, 2018). The second widely used component is automatic passenger counting (APC). This component typically relies on estimation techniques based on door loop counts or weight sensors. Especially, combining AVL and APC is promising as APC provides accurate timing of when a bus stops at a transit stop, which AVL cannot provide. AVL data analysis is can provide the means for evaluating a schedule plan's reliability. For example, by identifying route segments where greater schedule deviations, and therefore, the schedule plan should be adjusted by changing the timetable or by introducing bus priority lanes. Furthermore, data gathered using AVL is an often used data source by predictive analytics (Moreira-Matias *et al.*, 2015). But one has to consider that AVL data is often noisy as GPS accuracy is not always sufficient and therefore needs to processed and map-matched (Sun *et al.*, 2018). A process of positioning the coordinate of the point obtained from the tracking device into the bus network onto the street (Čelan and Lep, 2017). This is usually a computational expensive process. Besides AVL and APC data, static GTFS in combination with real-time GTFS can also be used to predict travel time, as often real-time GTFS also contains GPS information derived by AVL systems (Sun *et al.*, 2018).

Currently, much research has been done in the recent years on how predictive analytics can be applied to solve the travel time prediction problem in public bus transportation. In general, current research tries to answer the following question: Given the current position of a bus, when does the bus arrive at the next or one of the subsequent bus stations. To solve this problem different algorithms and approaches have been evaluated so far. Maybe the naivest approach is what is called the historical average travel time model, which is presented in the following section (Gurmu and Fan, 2014; Oruganti *et al.*, 2016; Čelan and Lep, 2017; As and Mine, 2018). This model relies on the availability of accurate AVL data. Once processed and bus arrival times for each bus at each stop have been calculated, one can build the historical average travel time model. However, numerous parameters affect travel velocity of buses such as density of traffic flow, administrative limitations, number of passengers or weather situation (Čelan and Lep, 2017). In the following section different models and approaches are discussed in more detail.

2.2.2 Bus travel time prediction using non-learning algorithms

There are multiple approaches and techniques to solve the travel time prediction problem of a bus. Probably the naivest model is the historical average travel time model, a non-learning method. This model is most often based on the processing and analysis of historical AVL data (Jeong and Rilett, 2004). Once the historical arrival times for each stop of a specific bus have been derived, one can calculate the travel time the bus needed between two adjacent bus stops. By aggregation according to different time periods it allows us to calculate the averaged travel time for a bus route segment. Consider a bus k, for which the model needs to predict the travel-time T for the route segment s (segment between stop i and downstream stop j). Each day is partitioned in eight time periods, $p \in (p1, p2, .., p8)$. Then the average travel time T for the route segment s at time period p can be calculated as:

$$\boldsymbol{T_s^p = A_j^p - (A_i^p + D_i^p)} \qquad \textbf{(1)}$$

Where A_j^p is the averaged arrival time of bus k at station j at period p. A_i^p is the averaged arrival time of bus k at station i and D_i^p is the averaged dwell time of bus k at station i. The average travel time T_s^p can now be used as a predictor to estimate future arrival times for bus $k + 1$ station j (Jeong and Rilett, 2004).

Čelan and Lep (2017) further developed this approach by analysing historical AVL data for a specific bus route in Maribor, Slovenia. By analysing the data, they found regular patterns for the bus's hourly averaged velocity. Buses have greatest average velocities early in the morning, late at night and during the weekend. Based on these findings they defined different time periods together with the corresponding specific average velocity. In order to accurately predict the arrival time of the bus at the stop requires the bus route trajectory. For this, Čelan and Lep (2017) defined different bus network model. Basically, the route trajectory was split in different nodes and links. An intuitive model they used, is to model bus stops and potential barriers such as roundabouts or crossings as nodes and links connect adjacent nodes. As they could locate the nodes, it is possible to calculate the distance for each link. Now, combined with an average velocity for a specific time period it is possible to estimate the bus travel time for each link given predefined time periods. Consequently, it is possible to predict the arrival time of the bus by extrapolating the arrival time using the equation:

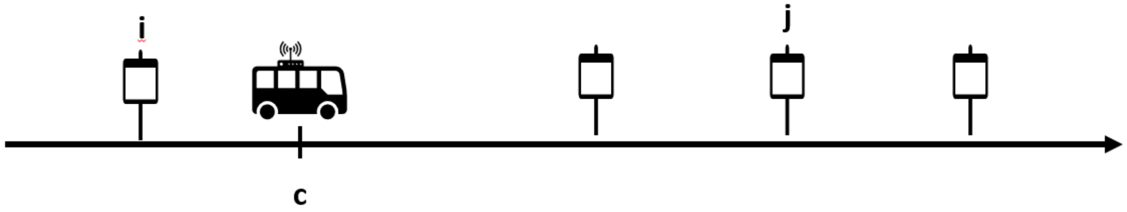
$$\mathbf{T}^{\boldsymbol{p}} = \boldsymbol{t}_{\boldsymbol{Lc}}^{\boldsymbol{p}} \cdot (\mathbf{1} - \boldsymbol{n}_{\boldsymbol{Lc}}) + \sum_{\boldsymbol{x}=\,\boldsymbol{L}_{\boldsymbol{c}+\mathbf{1}}}^{\boldsymbol{L}_{\boldsymbol{s}-\mathbf{1}}} \boldsymbol{t}_{\boldsymbol{x}}^{\boldsymbol{p}} \qquad \textbf{(2)}$$

Where T^{p} is the predicted bus travel time in the time period p from the current location to the location of the target bus stop. It is derived by summing up the travel times for each link in the model the bus needs to pass until the arrival of the target bus stop. t_{Lc}^{p} is travel time that the bus in period p uses link Lc within the bus network. It is multiplied by the percentage of travel time that remains for the bus to cover until the end of the current link. Within the sum sign the travel times (in time period p) for all links are summed up, which the bus needs to absolve to reach the target node. Using this approach, they were able to predict the arrival time for all following stations with a mean absolute percentage error (MAPE) between 13.3-16.5% depending on the network model. They achieved the best results when not modelling the bus station and potential barriers as nodes in the network model. But by defining the half-distance between two stops respectively potential barriers as nodes (Čelan and Lep, 2017). However, in general historical average models are only reliable when the traffic pattern in the area of interest is relatively stable or where congestion is minimal, e.g. in rural areas (Gurmu and Fan, 2014).

2.2.3. Bus travel time prediction using learning algorithms

Travel time prediction using artificial neural networks for regression

An approach using ANN to predict the arrival of a bus at a specific station was proposed by Gurmu and Fan (2014). Based on the current position of a bus, derived by AVL, they predict the arrival at a subsequent station using an ANN. ANN is a black box-type function that only provides an output and not a relationship between the independent variables and the target variable (Moreira-Matias *et al.*, 2015). Gurmu and Fan (2014) argue by using an artificial neural network nonlinear correlation between travel times can be captured to predict bus travel time at subsequent bus stops. As input variables they choose the current bus location c, the last served station i, the arrival station to be predicted j and the time of day.



The predicted arrival time $\boldsymbol{T^{cj}}$ for the $\boldsymbol{k}$ to the station $\boldsymbol{j}$ was calculated as:

$$\boldsymbol{T^{cj}} = \boldsymbol{T^{ij}} - \boldsymbol{T^{ic}} \qquad \textbf{(3)}$$

Where $\boldsymbol{T^{ic}}$ is derived by deducting departure time at stop $\boldsymbol{i}$ from the current time, at point $\boldsymbol{c}$. $\boldsymbol{T^{ij}}$ is the parameter predicted by the ANN and corresponds to the predicted travel time between stop i and stop $\boldsymbol{j}$. By deducting $\boldsymbol{T^{ic}}$ from $\boldsymbol{T^{ij}}$ it is possible to calculate the predicted travel time $\boldsymbol{T^{cj}}$ between location $\boldsymbol{c}$ and stop $\boldsymbol{j}$. Gurmu and Fan (2014) compared this approach to a historical average model similar to Čelan and Lep (2017) using the same data. Their results show that overall prediction accuracy has benefitted by using an ANN.

These two models were both evaluated by predicting travel time and hence calculating arrival for one specific bus route within a city. However, certainly these models could also be applied on a whole bus network. This has been demonstrated by the work of As and Mine (2018). Similar to Čelan and Lep (2017) they divided a day into eight time periods. They analyzed historical bus arrival data of one month and calculated for each adjacent station pair the averaged travel time within each time period. With this information they are able to extrapolate the arrival time of a bus at a following bus station within its route and clarify the variability of bus travel time over each time period. In addition, the proposed model by As and Mine (2018) contributed to the current research by the following two aspects. First, instead of solely using historical average time or ANN they predicted travel time with a time series-based approach. Therefore, they used a special form of an ANN, a so-called nonlinear autoregressive network with exogenous input (NARX) model. Especially dynamic nonlinear systems with time-series characteristics can be modelled by using NARX models. In general, the basic idea is to create a more dynamical model. As Gurmu and Fan (2014) they used historical average travel time for specific time periods as input variables for NARX model. The second contribution was by introducing a more dynamical input variable. By using a time series-based approach their model dynamically recalculates the historical average travel time for the period just before the current one. But by approaching the problem as a time-series their model could dynamically recalculate the historical average travel time for the time period just before the current one. This allows the model to capture information that is timely closer to the event it needs to predict. However, contrary to Gurmu and Fan (2014) this approach did not consider the current position of the bus as $\boldsymbol{T^{ic}}$ is unknown in the model proposed by As and Mine (2018). As and Mine (2018) directly compared the dynamical model with a static model, which also used NARX but without considering the average travel time for the preceding period. The results show that the dynamical model predicted travel time more accurate especially during the mornings where we could expect much traffic.

Travel time prediction using linear regression and Random Forest Regression

Besides ANN other regression algorithms have also been applied to solve the travel time prediction problem. Most common is the application of random forest models and multivariate linear regression (Moreira-Matias *et al.*, 2015; Oruganti *et al.*, 2016; Yamaguchi, As and Mine, 2019). Regression models have especially been applied to study the effects of different factors on the travel time. For example, it has been studied how weather affects bus travel time and how much these findings can be used to predict bus travel time in advance (Oruganti *et al.*, 2016). Approaching travel time prediction as a regression problem offers the possibility to evaluate the different factors that could have effects on travel time. On the one side, this would result in a deeper understanding in which factors are relevant to predict travel time. On the other side, it can be a valuable feedback for public transportation agencies to know which factors or even which combination of factors can lead to longer travel time and thus to delays in the transportation system. Random Forests are an ensemble learning method for classification and regression where a number of decision trees are constructed. For regression tasks the outcome variable is fitted for a regression model using each predictor. For each prediction the data is split at split points and the sum of squared error at each split is evaluated. The predictor resulting in in the minimum sum of squared error is selected for the node. Therefore, the underlying principle is that a group of weak learners can be combined to for a strong learner (Oruganti *et al.*, 2016). Oruganti et al. (2016) proposed a model to predict bus travel time based on multiple different variables including weather data. For the prediction model they used real-time transit feeds, which represent real-time updates of transit fleet information. It contains information about trip updates, service alerts and the current vehicle position. Furthermore, they used historical data about accurate bus's arrival and departure times. The purpose of their research was to study the effect of weather and other factors such as traffic on the transportation system. In order to achieve this, they included traffic flow information to their model in form of the current traffic-speed on the bus route. Furthermore, they included weather data such as temperature, wind speed, precipitation and other weather-related information to their model. Based on these variables their model is able to explain more than 70% of the variance in the bus travel time and their linear regression model is able to make future out of the box predictions with an out-of-sample error of 4.8 minutes, given information on bus schedule, weather and traffic. In addition, Oruganti et al. (2016) evaluated how well a linear regression model performs against a random forest model using the same data and input variables. The random forest model outperformed the linear regression slightly in terms of goodness fit, measured in R^2 = 73% for random forest against R^2 = 71% for linear regression. However, in terms predictive accuracy linear regression performed better than random forest (Oruganti *et al.*, 2016). These findings correspond to the ones outlined by Yamaguchi et al. (2019). They compared different machine learning algorithms to predict travel time. To the authors surprise, linear regression performed as well as the ANN model to predict travel time over time intervals as proposed by As and Mine (2018). Overall, Yamaguchi et al. (2019) could find that a Gradient Boosting Decision Trees model performed best in terms of MAE (mean average error) and RMSE (root mean square error).

On the other side the linear regression model enabled Oruganti et al. (2016) to determine the influence of different variables on the predicted outcome. For example it has been showed that in Nashville precipitation was not considered as a significant predictor, rather visibility and wind related factors were more important within a travel time prediction model using linear regression and Random Forest regression (Oruganti *et al.*, 2016).

In this section different approaches and techniques were presented. They have in common that they are all based on historical data analysis and predictive analytics to predict travel time for buses and calculate bus arrival times. The purpose of these papers is to evaluate the performance and prediction accuracy of the proposed approaches and techniques. However, in general these methods can be integrated in large-scale advanced public transportation systems as proposed by Chen et al. (2004) or Sun et al. (2018) and others (Shalaby and Farhan, 2003; Oruganti *et al.*, 2016; Gal *et al.*, 2017). On the one hand, in contrast to the models presented above these large-scale advanced public transportation systems use a more holistic approach to predict travel time or arrival delays. This approach enables to retrieve further input variables that can be used for prediction. On the other hand, this holistic approach to transportation systems results in more complex system-architecture of the prediction model. Often such models as discussed above are only on piece of the whole advanced public transportation system.

In the following different papers outline the full architecture of operational prediction systems for real-time bus arrival prediction and include sometimes multiple data sources and different techniques.

2.2.4. Architectures of advanced public bus transportation systems

The purpose of operational prediction systems, respectively advanced public transportation systems, is to predict the travel time of buses for downstream stops along a bus-service. The aim is to provide accurate bus arrival time information to the passengers in real-time. With accurate arrival information, transit users might efficiently schedule their departure time or adjust their itinerary according to the current transportation situation (Chen *et al.*, 2004). As these system aim to provide information in real-time, they often integrate real-time information collected by AVL and APC and data from other sources, that could affect travel-time such as weather (Oruganti *et al.*, 2016). In contrast, to the models outlined in the previous section, the systems discussed in this section are more overall solutions for advanced public transportation systems that would integrate such models as discussed in the previous section and combine it with other components.

Multi-component system

Chen et al. (2004) proposed a dynamic bus-arrival time prediction model based on the integration of weather information, historical APC data and individual on APC data. The bus-arrival time prediction consists of two major components. First, an ANN model that is based on historical trip data collected by APC units. The ANN model predicts travel time for each segment along the bus route, given trip starting time, day-of-week and weather conditions. While new generated trip data are added into the database regularly, training can be reconducted in order to ensure that the ANN model is up to date. Nevertheless, at the time the current ANN models did not encounter dynamic variables as the NARX model proposed by As and Mine (2018) and therefore could not adjust prediction using the most recent, real-time information for a bus trip. Therefore, Chen et al. (2004) integrated a second component to their prediction system. The second component is based on Kalman filter technique, which adjusts the predicted bus travel time by the ANN by accounting real-time information transmitted by the on-board APC unit of the bus in question. This allows to adjust online the travel/arrival-time prediction by accounting the most recent travel or arrival-time information. Shalaby and Farhan (2003) outlined a similar operational prediction system. Besides, APC data they integrate AVL data in their model as well. In addition to that they proposed a real-time user-interface for transit controller to assess the effect of bus expressing at one or more downstream bus stops or the effects of prolonging dwell time at one or multiple stations.

Embedding the snapshot principle

More recent proposed systems embed the snapshot principle in order to improve travel time prediction in operational prediction systems (Gal *et al.*, 2017; Sun *et al.*, 2018). The snapshot principle, is originally based in queueing theory and has evolved to enhance delay prediction in service processes (Senderovich *et al.*, 2014). It says that considering a queuing process, an adequate delay prediction for a newly enqueued customer would the delay of either the last customer to enter the service or the delay of the customer at the head of the line (Senderovich *et al.*, 2014). In transportation services the snapshot principle can be applied as follow. Consider a bus route $\boldsymbol{r}$, with $\boldsymbol{k}$ representing the current bus-service. The aim is to predict travel time for route segment $\boldsymbol{s}$ between two adjacent bus stops ($\boldsymbol{s_i}$ and $\boldsymbol{s_j}$) within the route $\boldsymbol{r}$. The snapshot principle suggests using the travel of the previous bus-service $\boldsymbol{k-1}$, which passed segment $\boldsymbol{s}$ as a travel time prediction. Therefore, the travel time of bus $\boldsymbol{k-1}$ can be interpreted as an indicator for the current traffic situation and can be used as travel-time approximation for bus-service $\boldsymbol{k}$. Because it is likely that the following bus $\boldsymbol{k}$ encounters similar conditions as the preceding $\boldsymbol{k-1}$. Nevertheless, scheduled bus transportation services are often repeatedly executed after specific time intervals, for example every hour. However, the traffic situation might change during this time interval and using the travel time of $\boldsymbol{k-1}$ as an approximation might be insufficient (Gal *et al.*, 2017; Sun *et al.*, 2018). Therefore to achieve better travel time approximation for bus $\boldsymbol{k}$, Sun et al. (2018) proposed to identify route segments that are shared by multiple bus routes. This is the case if segment $\boldsymbol{s}$ is served by multiple bus routes. By identification of shared route segments, the time interval between two bus-services of different bus-routes travelling through $\boldsymbol{s}$ can be minimized, resulting in better travel time approximations.

Since bus delays are often induced by car traffic, it is tempting to use information retrieved by the snapshot principle and use it as a baseline for travel time prediction. The basic idea is to embed most current information to predict travel time. This idea has also been taken up by the proposed NARX model of As and Mine (2018) and by approaching the problem as a time-series. But, in contrast to Sun et al. (2018) their model only respects an average travel time calculated within a preceding timeframe, which is not as specific as the travel time of the directly derived by the preceding bus. On the other side, Gal et al. (2017) have shown that embedding snapshot information as input variables in regression trees can enhance prediction accuracy compared to the same model without snapshot information.

2.3 Delay prediction in railway networks

During 2006-2007 the British national rail network registered 800’000 delays, which led to 14 million train-minutes delay, which in turn would cost the passengers about £1 billion in lost time (Lessan, Fu and Wen, 2019). Therefore, reducing the delays is of great importance for railway operators but is also of interest for the economical growth of a region or a whole nation. Furthermore, similar as for bus networks train delay prediction aims to provide useful information to traffic management and dispatching processes through the usage od state-of-the-art tools and techniques (Oneto *et al.*, 2018). Delays in railway networks can have different causes, such as disruptions in the operations flow, accidents, malfunctioning or damaged equipment (Oneto *et al.*, 2018).

2.3.1 Available data sources and general problem setting

Railways are among the industries in which the application of big data analytics is a topic of big interest. Big data analytics have been revolutionizing the railway industry by contributing to decision-making processes within railway companies. This development has been engaged by collecting data from different sources within railway operations (Ghofrani *et al.*, 2018). The major data source for delay prediction and delay analysis is train describer data. Train describer systems identifies a train at a particular position keeps track of every movement. The train describer system successively registers this movements as events on a route of train line and writes these events to log files with the corresponding time (Goverde and Hansen, 2000). It is worth noting the exact position of the train is not contained in train describer data. The location is assessed by the traffic control signals, whose location is known, and which registers incoming and outgoing trains for a specific track segment. This signal is then transmitted to the train describer system. Originally these logs have been kept only for few days to support investigation of possible incidents (Goverde and Hansen, 2000). However, since companies realized that this data can be used to evaluate timetable performance and providing insight in railway operations the importance of this data for railway companies has risen. Collecting, processing and transforming these train describer record data in combination with timetable data for descriptive analysis of train delays and timetable improvements can be seen as the first application of big data in railway operations (Ghofrani *et al.*, 2018). Similar to AVL and APC data sources in bus transportation systems, the availability of train describer data enabled to perform big data analytics and enhance railway transportation systems in different ways.

Ticket sales are another interesting data source that can be embedded in predictive analytics for delay prediction. Ticket sales data has been used in combination with delay data in order to examine the impact of lateness on demand (Batley, Dargay and Wardman, 2011). In contrast to ticket sales data with smart cart data it is possible to retrieve the passenger's destination. This in turn allows to build route choice models and consequently to derive passenger punctuality instead of train punctuality (Ghofrani *et al.*, 2018). This gives train-dispatcher a new evidence-based basis for decision-making.

In contrast to the available data sources in bus transportation networks, namely AVL and APC used for delay prediction, data sourced from train describer has some crucial advantages. As mentioned by Sun et al. (2018), accurate bus arrival and departure data is not always available, especially in real-time. Furthermore, for many bus transportation systems APC is missing and therefore it is not possible to provide accurate timing of when a bus's arrival (Sun *et al.*, 2018). On the other hand, if AVL is missing it is not possible to locate the bus. In addition, the real-time systems often have many problems due to reasons, such as low networking bandwidth and delays in upload which results often in noisy GPS position data (Sun *et al.*, 2018). Such problems with noisy data and missing components seem not be an issue for railway transportation networks. Still train describer data suffers from transmission delays or noise, but these are usually only fraction of a second and is considered as negligible. Furthermore, missing values in train describer data are encountered by applying different logical rules (Goverde and Hansen, 2000).

Due to the high accuracy of the available data, delay prediction in railway network has more focussed on predicting the arrival delay instead of the travel time between to stations (Peters *et al.*, 2005; Yaghini, Khoshraftar and Seyedabadi, 2013; Marković *et al.*, 2015; Oneto *et al.*, 2016, 2017). Furthermore, in highly scheduled railway networks delays can cause domino-effects an therefore affect the whole network (Goverde, 2010). For delay prediction in bus transportation systems, most used algorithms are machine learning algorithms. However, delay prediction in railway transportation systems makes also use of other algorithms such as stochastic-graph models (Berger *et al.*, 2011) or timed event graph models (Hansen, Goverde and Van Der Meer, 2010).

In the next section different train delay prediction models will be discussed. First part is dedicated to models that have been developed as part of rail traffic management systems. The second part will focus on models, which use different machine learning algorithms.

2.3.2 Train delay prediction using graph-based models

A common approach to predict train delays in railway network is the use of graphical representations of the railway network. (de Fabris, Longo and Medeossi, 2008; Hansen, Goverde and Van Der Meer, 2010; Berger *et al.*, 2011; Kecman and Goverde, 2015; Ghofrani *et al.*, 2018). Using this graphical modelling approach Goverde (2010) presented an effective way to propagate delays over a whole railway transportation network, which is crucial to understand the domino effect of secondary delays a single delayed train may cause over the entire network due to train connections and route conflicts. In addition, graphical models can be directly integrated into the railway traffic system as suggested by Kecman and Goverde (2015) or Berger et al. (2011). By embedding them directly into the railway traffic system the model can make use of an incoming stream of data in real-time. In general Kecman and Goverde (2015) suggested to classify such models into microscopic and macroscopic models. Macroscopic models such as proposed by Berger et al. (2011) focus only on station events such as arrivals and departures at stations. Microscopic models on the other side also include the prediction of signal events. Signal points can be referred as reference points between train stations. These signal points register the entrance and exit of a train for a particular track segment.

Microscopic modelling approach

The framework proposed by Kecman and Goverde (2015) suggests, that the traffic control system continuously provides route and connections plans of the trains within the railway network. In addition, the actual traffic state, including the current train positions and delays is continuously provided by a monitoring system. The prediction model is based on a directed acyclic graph with dynamic arc weights. The graph topology is defined by the actual process plan, including train orders, routes and connection plan as well as current train positions within the railway network. Therefore, a change of the actual plans, such as changing the relative order of trains, adding or cancelling trains or modifying routes results in an update of the graph topology. Based on the actual traffic state, comprising of current train positions and delays the prediction model predicts event times for each node in the graph. Nodes represent arrival and departure events or other signal points within a railway network. Therefore, a train route is represented as a sequence of track sections and signals. The predicted delay of a train for a node can be obtained by subtracting the predicted event times from the scheduled event times. The prediction of event times is based on max-plus algebra. Max-plus algebra is a discrete algebraic system, which allows to represent the behaviour of a class of discrete event systems by simple linear equations. This equation can be used to realize modelling, analysis and control of a system (Goto, 2014). The arc weights represent the estimated process times that are computed based on the actual traffic state and processed historical data in order to account running time variations depending on the current delay and peak hours. To improve prediction accuracy during deployment the monitoring system gives feedback of the realized process times, which in turn are included for the dynamic arc weights calculation. Using this feedback mechanism, the accuracy of the prediction model could be significantly improved.

The advantage of this model is that it allows to define a prediction horizon. The prediction horizon describes the time window in future, for which all events are predicted. In their model, Kecman and Goverde (2015) could register a drop in prediction accuracy for increasing time horizons for up to 120min. Another advantage of Kecman and Goverde's model is that it respects train-collision rules, defined by the rail traffic management system. This means that the model would not predict arrival times for a train that would conflict those of another train. This is ensured by highlighting the graph arcs that are used by multiple trains. The disadvantage of this approach for predicting train delays is the need of accurate railway process plans, which include track occupation information, running orders, connections plan and timetable information. Furthermore, the signal point's location must be available together with their delivering incoming stream of data. Most often these railway process plans, and the railway infrastructural data is only accessible for the railway network operators and are not open to public. Therefore, this approach can hardly be used by non-partnering institutions. Another downside of the dependence of this information is that the resulting graphical model must be completely updated if the process plans are adjusted or changed, which is computationally expensive (Lessan, Fu and Wen, 2019). Another uncertainty in this approach is how such a model performs in terms of accuracy if the density of signal points and train stations is low within the rail network. If we assume that the probability of a delay-causing incident increases with the spatial distance between signal points and train stations. Then, one could expect less accuracy for the proposed model if the density of stations and signal points is low within the network. As the proposed framework was evaluated solely for a busy corridor of a railway network in the Netherlands, it is unknown how this framework performs over an entire railway network.

Macroscopic modelling approach

Berger et al. (2011) presented a stochastic delay prediction model using a graph-based representation of the railway network. Their model is able to propagate delays over the rail network and forecasts arrival and departure events. The graph models the train schedule and the waiting conditions between planned transfer possibilities for the whole railway network. The proposed model is formulated with respect to an event graph, which is directed and acyclic to allow delay propagation in a topological order of events. The model uses a discrete distribution of driving time profiles on travel arcs which depend on the departure time, but also on train category or track conditions. Further, Berger et al. (2011) define waiting policies for each train by defining the maximum amount of time a train waits at a station until another train arrives at the same station to ensure passenger transfers. A train that has a transfer-relation with another train is called a feeder train. These waiting policies are defined for any pair of arriving and departing trains for which a transfer arc is defined in the railway operations plan provided by the railway agency. Hence, new transfer possibilities due to other delayed trains are not reflected and would complicate the implementation of the model (Berger *et al.*, 2011). Further the model relies on several basic assumption. It is assumed that a train can arrive at any time after the planned arrival or departure time. This is also incorporated in the prediction model in order to propagate delay over the network for arrivals and departure that lie in more distant future. Another important assumption is that the distributions of arrival times of all feeder trains of a given train are stochastically independent. This assumption might simplify the model and enable fast delay computation over the whole network. But on the other side it neglects the inherent interdependencies between trains in combination with infrastructural limits, for example track bottlenecks in front of stations.

However, simulations with several distributions of travel times on travel arcs results in interesting insights into the robustness of the planned schedule against small fluctuations. A pivotal asset of the proposed model by Berger et al. (2011) is, that it integrates with a stream of online messages about the delay status of trains from the railway company, which corresponds in general to train describer data. This allows to immediately propagate these messages through the whole railway network and compute to its impact on future arrivals and departures in the near and more distant future. In order to assess the quality of model they compared it to realized data provided by the Deutsche Bahn AG. For this they implemented the German timetable of 2011 and the waiting policies defined by the railway operating agency. The implemented model is able to complete stochastic delay propagation of a whole day within 14 seconds over the whole German railway network. The mean absolute error ranges between 4 to 6min depending on prediction horizon. For example, for a prediction that lies in 120min in the future the mean absolute error lies between 4 to 5.5min depending on day and the implementation of waiting rules. Compared to the results that were achieved by Kecman et al. (2015) (40sec with 120min prediction horizon) the average error is much higher but so is the number of predicted events, as the prediction involved the entire railway network instead of a highly frequented corridor.

2.3.3 Train delay prediction using learning models

Train delay prediction using ANN

The use of artificial neural networks for train delay prediction has been used widely and also directly compared to other machine-learning algorithms (Peters *et al.*, 2005; Yaghini, Khoshraftar and Seyedabadi, 2013; Marković *et al.*, 2015). Peters et al. (2005) proposed a train delay prediction model that is based on the principle of classical pattern matching. This means, that different delay situations of a train network result in new concrete constellations. Therefore, they suggested to train the ANN in way that different delay scenarios are represented by a particular input pattern for the artificial neural network. The output of the ANN corresponds to the predicted pattern of the input delay scenario. The architecture of the artificial neural network allows to predict the delay of the upstream or downstream trains based on the delays currently incurred in the network (Lessan, Fu and Wen, 2019).

Another approach is proposed by Yaghini et al. (2013) by classifying train delays into different bins according to the delay duration. In their model ANN was used to classify the delay based on different input variables, which is a supervised classification approach and therefore their model does not predict delay but rather a delay approximation. For this, they used passenger train delay data provided by the Iranian Railways, which containing all registered trains for several years, in total nearly 5.5 million trains. The dataset contains information about the date the train was registered, the origin and destination station and the railway corridor the train followed. At the same time these variables were used as input variables to predict the delay approximation of a specific train. Their findings show that the accuracy of the ANN model varies depending on the neural network architecture and the way the input data has been encoded and normalized. The classification model proposed by Yaghini et al. (2013), was evaluated against other machine-learning algorithms, namely decision trees and multinomial logistic regression, whereas the ANN-model achieved the best results with highest accuracy around 90% correct classified. Contrary to Yaghini et al. (2013), Marković et al. (2015) used ANN for regression to predict the arrival delay of passenger trains arriving at a particular station. In direct comparison with a Support Vector Regression (SVR) model they concluded that the proposed model using ANN outperformed the SVR model on the training data. However, the more relevant comparison of the two models on the test data indicates better generalization power for the SVR with achieving to explain 65% of the variance in train arrival delays for a particular station (Marković et al., 2015).

Delay prediction using regression models

Contrary to Yaghini et al. (2013) , Marković et al. (2015) used a regression approach to predict delay of passenger trains in Belgrad. Their research focused on predicting arrival delays for all trains arriving at a particular station within the Serbian railway network. The aim was to determine the underlying relational function between train delays and railway network properties. For this, they determined several factors that estimate influence of railway infrastructure on train arrival delays. These factors have been highlighted by discussions with experts. Based on this and the available dataset they selected the following variables to predict arrival delays for a station:

1. Passenger train category (nominal: suburban, regional, long-distance).
2. Scheduled time of arrival at station (continuous).
3. Infrastructure influence defined by expert opinions (ordinal: 3, ..., 9).
4. Percent of journey completed distance-wise (continuous).
5. Distance travelled (continuous).
6. Time travelled (continuous).
7. Headway (continuous).

The variable "infrastructure influence" was determined by an expert and was based on different aspects and is characterizes a train route. For example, how many stations, stops, junctions and crossings the train does complete on his journey. In addition, the percentage of single-track and restricted speed were determined for each route were considered among other factors by the expert to determine the influence of infrastructure to arrival delays. Using this variables Marković et al. (2015), were able to explain 58% of the variance in train arrival delays using the ANN algorithm for regression. As already mentioned above, the generalization power could be improved by using a SVR algorithm in favour of an ANN.

Another interesting approach to predict train arrival delays is by considering the influence of other delayed trains in the network as proposed by Wang and Work (2015). This approach follows the principle that if a train is delayed it influences other closely scheduled trains, hence these trains might experience so called knock-on delays. As a base the model by Wang and Work (2015) relies on a regression model using historical data, assuming that delays from one trip to the next follow an vector autoregressive process. This is very similar to the historical average model discussed in section 2.3.3. However, instead of using the average as a predictor, Wang and Work (2015) propose to use the historical delay data of all previous trips as variables within a regression model, which in turn predicts all arrival delays for the following trip. In order to improve this approach, they propose to predict arrival delay along the current trip using delay-information of the previous train arriving at the stations and also to use delay information of other trains that share the same corridor to capture knock-on delays. More in detail, the input variables used in the regression model to predict the arrival delay $\boldsymbol{dy}$ for a train $\boldsymbol{k}$ at station $\boldsymbol{j}$ is are:

1. Delays of $\boldsymbol{k}$ on previous trips
2. Delays of $\boldsymbol{k}$ at previous stations on current trip
3. Delays of trains $\boldsymbol{Q}$ leaving from a neighboring station of station $\boldsymbol{j}$ within one hour

Their findings show that the inclusion of the third variable does not lead to a significant improvement of prediction accuracy compared to a regression model using the first two variables. Wang and Work (2015) conclude that once a train is delayed at a station, it is observed that the delay will propagate for several stations and therefore it is captured by the second variable. As a result, that if the train $\boldsymbol{k}$ has been delayed because of another train $\boldsymbol{q} \in \boldsymbol{Q}$ the resulting delay for $\boldsymbol{k}$ will be propagated along its trip and therefore is captured by the second variable again.

Nevertheless, the same principle has been taken up by the intensive research conducted by Oneto et al. (2016, 2017, 2018). In addition, Oneto et al. (2016) emphasized that current research does not take into account the variety of factors affecting railway operations, such as drivers behaviour, passenger volumes, weekday, holiday etc.. Instead by using advanced analytics algorithms it is possible to perform a multivariate analysis over data coming from different sources but related to the same phenomena. The idea behind this approach is that the more information is available for the creation of the mode, the better the performance will be (Oneto *et al.*, 2016). The proposed solution integrates multiple prediction models, which perform each a multivariate regression on a trains delay profile along its itinerary and other possible correlated variables, such as weather and information about other trains travelling on the network by using a time series and a macroscopical approach (Oneto *et al.*, 2016, 2017, 2018).

Advanced train delay prediction system using machine learning

As already mentioned an advanced delay prediction system using learning algorithms was described by Oneto et al. (2016, 2017, 2018). Advanced in this context refers to the extent that the proposed system covers. In contrast to other train arrival delay prediction models this approach predicts the arrival delays for all subsequent stations within a trip based on the current situation at time $\boldsymbol{t_0}$. The variables used to predict arrival delays for all subsequent stations for train $\boldsymbol{k}$ included are listed below (Oneto *et al.*, 2016):

1. Delays within timeframe $\boldsymbol{t_0 - \delta^-}$
2. Actual Running times within $\boldsymbol{t_0 - \delta^-}$
3. Dwell times within $\boldsymbol{t_0 - \delta^-}$
4. Weather condition within $\boldsymbol{t_0 - \delta^-}$
5. Above four variables of all trains running on network since $\boldsymbol{t_0 - \delta^-}$
6. Forecasted weather conditions for all subsequent stops

The variable weather condition summarizes different measurements related to weather, such as the atmospheric pressure, humidity, solar radiation, wind and rainfall. The timeframe $\boldsymbol{t_0 - \delta^-}$ has been set equal to the time in the timetable where $\boldsymbol{k}$ starts its trip. Further, each trip is characterized by a specific route, which is defined according to a specific sequence of stations. To account these characteristics a prediction model is built for each route. These models work together in order to make possible to estimate the train delay of a particular train during its entire trip (Oneto *et al.*, 2018). Furthermore, by including delays and running times of all other trains running in the network during $\boldsymbol{t_0 - \delta^-}$ the input variable scope has been shifted from a vehicle-perspective to a network-perspective. Consequently, the input variables used for delay prediction are not limited to trip-related characteristics, such as daytime, weekday, destination station etc., instead delay prediction accounts also time-dependent network-related characteristics. Oneto et al. (2018) argue that using this approach the actual distribution of the train delays in the railway network is captured and therefore the intrinsic time varying nature of the delay phenomenon on railway networks can be addressed and leads to increased performance.

The performance of the proposed methodology has been validated using train describer data provided by the Italian Infrastructure Manager that controls all the traffic of the Italian railway network. As a prediction algorithm they explored the use of deep extreme-learning machines. Extreme-Learning machines are a subtype of artificial neural networks, which are considered to provide good generalization performance at extremely fast learning speed (Huang, Zhu and Siew, 2006). The results presented show that the proposed model outperforms the prediction system that is currently in place in Italian railway operating system, which is similar to the model described by Kecman and Goverde (2015) in terms of accuracy. The model presented by Oneto et al. (2018) is able to predict train arrival delays for the subsequent station with a mean absolute error of 1.5min. Furthermore, Oneto et al. (2016, 2017, 2018) contributed to the research aim of identifying the underlying functional relation between network characteristics and arrival delay as proposed by Marković (2015).

2.4 Summary of travel-time and delay prediction in transportation networks

Current research within the field of bus transportation predictive analytics focuses on the travel time prediction rather than arrival delay prediction as in railway transportation. A major challenge for bus transportation network is to account traffic load from private transportation modes, which share the same infrastructure. Until now, most research has been conducted on predicting travel time on an operational level (Moreira-Matias *et al.*, 2015). In both transportation domains the aim is to build models, which perform efficiently and achieve high prediction accuracy to support scheduling task and provide real-time information to end-customers.

In general, we can distinguish between models using learning and non-learning algorithms. Non-learning models in bus transportation rely solely on historical data analysis to calculate a predictor, as shown by Čelan and Lep (2017). In contrast Kecman and Goverde (2015) and Berger et al. (2011) proposed non-learning dynamic micro- and macroscopic models to predict train arrival delays and their impacts on the transportation network. Nevertheless, such systems seem not applicable for bus transportation network, as the initial situation regarding infrastructure and network access cannot be compared. In contrast to railway operators, bus transportation agencies do often not have the same possibilities to control and monitor the corresponding infrastructure as in railway operation systems. Another aspect is that non-learning micro- and macroscopic models rely on modelling specific railway process plans. Therefore, these models have to be updated after each changes occur in process plans (Oneto *et al.*, 2016). Furthermore, it is stated that such models are considered as not adaptive enough to incorporate the domain knowledge of local dispatchers and networks' characteristics (Lessan, Fu and Wen, 2019)

Regarding learning algorithms different learning algorithms have been evaluated for travel time prediction (Peters *et al.*, 2005; Marković *et al.*, 2015; Wang and Work, 2015; Oneto *et al.*, 2016). Travel time prediction using ANN are often among those with the highest accuracy in bus transportation (Gurmu and Fan, 2014; Moreira-Matias *et al.*, 2015; As and Mine, 2018). However, the underlying input-output function the ANN models uses for predicting a value is unknown. Contrary, Oruganti et al. (2016) showcase how other regression methods can be used with high accuracy and in surplus offer the possibility for interpretation. A similar approach was also proposed using SVR, which even outperformed ANN on the same data regarding generalization accuracy (Marković *et al.*, 2015). In railway transportation learning algorithms are most often used within regression tasks to predict arrival delay in railway networks (Marković *et al.*, 2015; Wang and Work, 2015; Ghofrani *et al.*, 2018; Oneto *et al.*, 2018). Supervised classification to predict delay approximations have also been evaluated (Yaghini, Khoshraftar and Seyedabadi, 2013; Lessan, Fu and Wen, 2019). A problematic aspect using a supervised classification approach is that most classification algorithms assume normal-distribution of the data (Sun, Wong and Kamel, 2009). However, it has been found that train delays typically follow a Gamma, Weissbull or exponential distribution (Yuan, Goverde and Hansen, 2006; Marković *et al.*, 2015). Hence discretization of delay and the preparation of training data for the algorithm should be considered carefully to avoid overfitting.

More recently, the research has focused in both transportation domains to evaluate and develop systems based on a more holistic approach to the network. The focus for input variables used in the prediction system has shifted from trip-related towards trip- and network-related. As one can see in the railway transportation domain several attempts exist to model and assess the effects of delays and its propagation over the whole network. For this, interactions between trains leading to delays have been considered (Wang and Work, 2015; Oneto *et al.*, 2018). In bus transportation networks, embedding the snapshot-principle into prediction systems has been showcased as an effective mean to approach the travel time prediction problem from a more holistic network-perspective (Gal *et al.*, 2017; Sun *et al.*, 2018).

2.5 Research gap and research question

Current research has been focusing on the design of complete transportation delay prediction systems. The aim is to build transportation delay prediction systems that are fast and accurate. This is a crucial requirement in order to deploy such systems within scheduling tasks on an operational and tactical level (Berger *et al.*, 2011; Kecman and Goverde, 2015; Gal *et al.*, 2017; Oneto *et al.*, 2018; Sun *et al.*, 2018). As emphasized in the previous section, for train arrival delay prediction using machine-learning algorithms, the scope for input variables has shifted from trip-related to trip- and network-related variables. Using train describer data, trip-related variables are the most obvious variables to extract, as train describer data captures includes the relevant information to identify a trip. On the other hand, network-related variables are already more sophisticated. The extraction of network- related variables requires a deepened understanding in data structure and how different trips are related to each other. For example, to extract variables based on the snapshot principle one has to identify different routes and the corresponding trip segment. In general, current research has focused on improving delay prediction by exploring different algorithms and also by including different input variables for train delay prediction. However, the impact of network-related input features and other features on the prediction accuracy in the models has been explored only marginally. This might come hand in hand with the use of very sophisticated machine-learning algorithms as ANN and SVR, which do not provide any means to explore the contribution of different features to train arrival prediction. In general, the main objective of this thesis is to investigate on analysing the input feature space used in machine-learning based train delay arrival prediction. In order to build improved prediction models, it is necessary to have a better understanding, which kind of input features contribute to train arrival delay prediction, which leads us to the first research question.

Research question 1: How do different feature categories, as being used in existing literature, contribute to train arrival delay prediction?

Approach: To investigate this question, a train delay prediction system according to the state of the art is built using a similar approach as proposed by Oneto et al. (2018). For this the input features will be categorized to according to their purpose. Afterwards different combinations of input features categories will be used to train the prediction model in order to evaluate them against each other. Furthermore, instead of using a machine-learning algorithm that hardly allows any investigation on the input features contribution, a gradient boosted regression tree algorithm will be used. For this purpose, the use of a tree-based algorithm is senseful, because it allows to analyze the feature importance of each feature used for the prediction (James *et al.*, 2013). Consequently, the feature importance's of the different input feature can be analysed and categorized.

Hypotheses 1: In railway transportation, results by Wang and Work (2015) indicate that the influence of other trains in the network is negligible. Therefore, it is expected that this would be reflected in the Gini Index value of the corresponding features. Further, it can be expected that trip-related variables contribute to the prediction of train arrival delays and therefore should be reflected by the feature importance value of the variable.

The second research question relies on the notion that public transportation systems are containing the nature of physical networks as they include the combination of lines and nodes that intersect with each other. This network nature of transit systems has been widely studied in the past (Derrible and Kennedy, 2011). Derrible and Kennedy (2011) emphasized that the field of graph-theory appears particular fitted to address problems of network design in public transportation. Centrality measures within this field have been used to identify the topological characteristics of railway and metro systems (Derrible, 2012; Tu, 2013; To, 2015). The centrality characteristics analysis is useful for the railway transportation management and operation affairs and is the basic for the network's vulnerability analysis (Tu, 2013). Furthermore, Lee et al. (2016) highlighted that timetable quality is one among various factors that can cause train arrival delays. In fact, timetable determines how well capacity is utilized and how stable the operations are within a railway network (Sameni, Landex and Preston, 2011). Consequently, one can argue that variables related to timetable should be included for the prediction of arrival delays, as they may contribute to identify the underlying relational function between railway network characteristics and train arrival delays. However, to the authors knowledge this conceptual source to engineer a new category of input features for train arrival delay prediction has not been explored yet. This will be addressed by using different centrality measures to assess timetable quality, by approaching the timetable as a graph. The graph-approach allows to calculate topological properties of a station using centrality measures. These station-related features should reflect the underlying topological properties of a station within the current scheduled operations of a railway network. This way, hidden timetable instabilities related to a station that induces arrival delays could be detected by a machine-learning algorithm to predict train arrival delay.

Research question 2: How do input features capturing topological properties between stations contribute to train arrival delay prediction?

Approach: Based on timetable information a graph will be built, where railway stations are represented as nodes. A train trip serving two stations is represented as an edge that connects the two corresponding station nodes. All trips are characterized by their schedule times as defined in the timetable provided by the railway operator. This allows to filter the trips according to time, which results in a filtered graph view representing the topological relations between stations and trips for a specific timeframe. Based on this reduced timetable centrality measures will be calculated for each station to capture the topological properties of a station within the scheduled operational setting.

Hypotheses 2: It can be expected that station-related features have an impact for train arrival prediction. This should be represented in improved accuracy of the prediction model and reflected in the feature importance of the station-related features.

The third research gap that has been emphasized regarding train arrival delay predictions is concerned about data availability. Most often train-describer data is not open to public. However, if the opportunity arises to get access to this data, it is senseful to know how much data is needed to train a prediction model accurately in advance. This knowledge can be useful for negotiations with railway operators. Therefore, the third research question can be formulated as follow.

Research question 3: How does the prediction model perform differently with different amount of training data?

Approach: The train arrival delay prediction system will be trained by increasing data sizes steadily. After each step the model predicts arrival delays using a test dataset. In this way the accuracy of the model can be assessed after each increase in training data size.

Hypotheses 3: We can expect that with increasing training size the accuracy the predictions made by the train arrival prediction system will improve. Further it is expected that the

3. Study Area and Data availability

The backbone of the Swiss national public transport system is the “Fernverkehrs-Netz”, which consists of the long-distance railway traffic. The long-distance traffic concession is granted by the Swiss government to the Schweizerischen Bundesbahnen (SBB). The most central aim for the long-distance traffic is to connect all areas of action and superordinate centres of Switzerland and integrating Switzerland into the European major traffic axes (Bundesamt für Verkehr BAV, 2017). The long-distance railway network serves all regions within Switzerland and provides the reference pulse signal for the nation-wide pulse timetable (Schweizerische Bundesbahnen, 2017). Therefore, service disruptions of the long-distance traffic affect the whole public transportation network in Switzerland. This emphasizes the importance that the long-distance traffic is on schedule in order to minimalize negative effects on economy and society. This is the reason why this thesis will focus on predicting delay for all train services that are involved in the long-distance traffic. The motivation behind this decision is that if it is possible to predict delay accurately it might be also possible for the small-distance traffic in urban areas. As outlined above it is important that public trains operated by the SBB are on schedule to guarantee transit connections with other more regional operating public transport agencies. In a highly scheduled railway traffic network, a single delayed train may cause a domino effect of secondary delays over the entire network, which is a main concern to planners and dispatchers (Goverde, 2010).

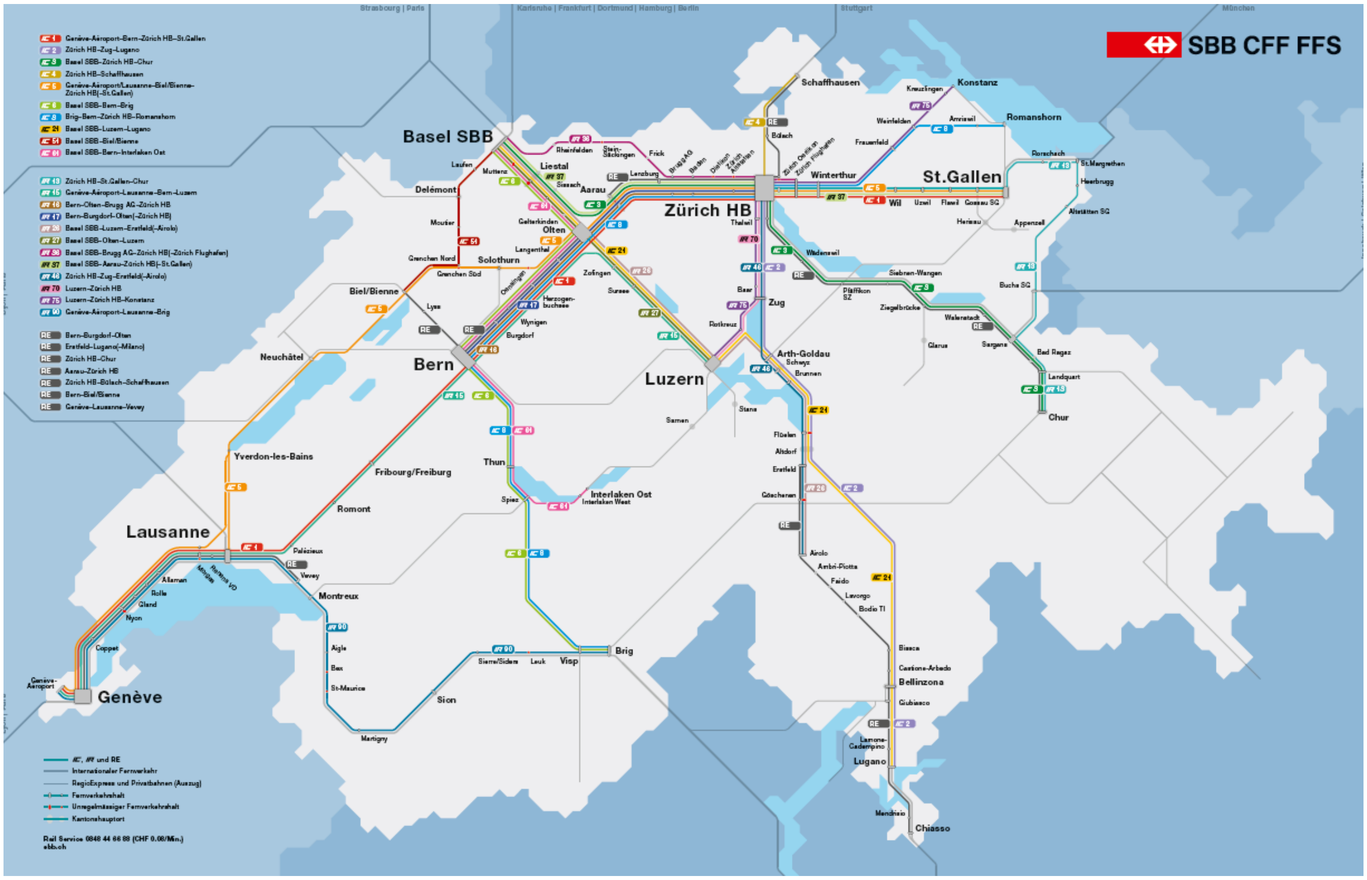
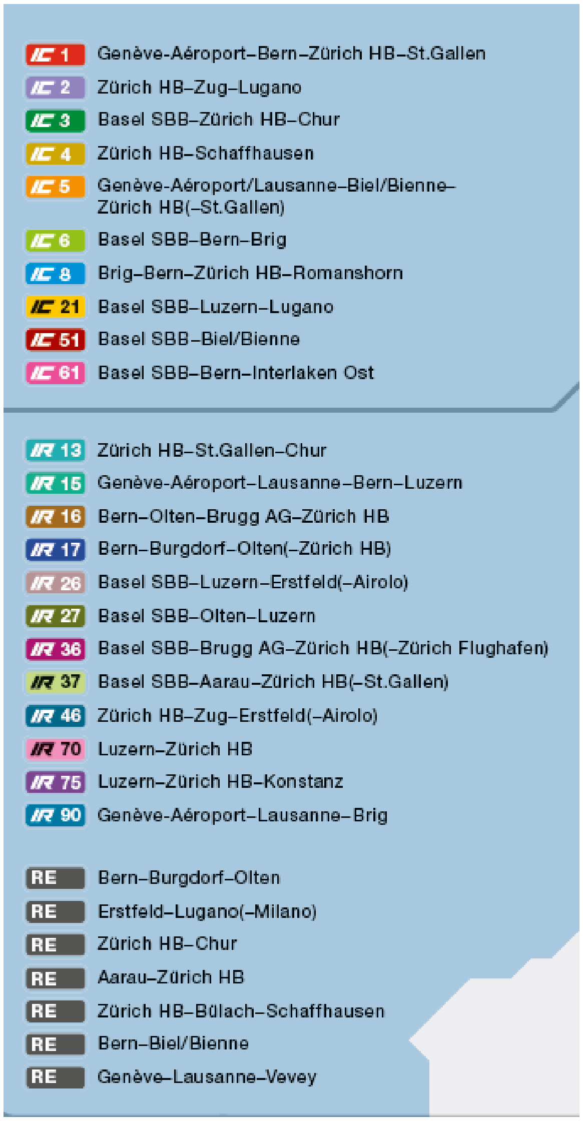

SBB CFF FFS
IC 1 Genève-Aéroport–Bern–Zürich HB–St.Gallen
IC 2 Zürich HB–Zug–Lugano
IC 3 Basel SBB–Zürich HB–Chur
IC 4 Zürich HB–Schaffhausen
IC 5 Genève-Aéroport/Lausanne–Biel/Bienne–Zürich HB(–St.Gallen)
IC 6 Basel SBB–Bern–Brig
IC 8 Brig–Bern–Zürich HB–Romanshorn
IC 21 Basel SBB–Luzern–Lugano
IC 51 Basel SBB–Biel/Bienne
IC 61 Basel SBB–Bern–Interlaken Ost
IR 13 Zürich HB–St.Gallen–Chur
IR 15 Genève-Aéroport–Lausanne–Bern–Luzern
IR 16 Bern–Olten–Brugg AG–Zürich HB
IR 17 Bern–Burgdorf–Olten(–Zürich HB)
IR 26 Basel SBB–Luzern–Erstfeld(–Airolo)
IR 27 Basel SBB–Olten–Luzern
IR 36 Basel SBB–Brugg AG–Zürich HB(–Zürich Flughafen)
IR 37 Basel SBB–Aarau–Zürich HB(–St.Gallen)
IR 46 Zürich HB–Zug–Erstfeld(–Airolo)
IR 70 Luzern–Zürich HB
IR 75 Luzern–Zürich HB–Konstanz
IR 90 Genève-Aéroport–Lausanne–Brig
RE Bern–Burgdorf–Olten
RE Erstfeld–Lugano(–Milano)
RE Zürich HB–Chur
RE Aarau–Zürich HB
RE Zürich HB–Bülach–Schaffhausen
RE Bern–Biel/Bienne
RE Genève–Lausanne–Vevey
IC, IR und RE
Internationaler Fernverkehr
RegioExpress und Privatbahnen (Auszug)
Fernverkehrshalt
Unregelmässiger Fernverkehrshalt
Kantonshauptort
Rail Service 0848 44 66 88 (CHF 0.08/Min.)
sbb.ch
Strasbourg | Paris
Karlsruhe | Frankfurt | Dortmund | Hamburg | Berlin
Stuttgart
München
Basel SBB
Zürich HB
St.Gallen
Schaffhausen
Konstanz
Romanshorn
Winterthur
Delémont
Olten
Aarau
Solothurn
Biel/Bienne
Neuchâtel
Bern
Luzern
Zug
Arth-Goldau
Chur
Yverdon-les-Bains
Fribourg/Freiburg
Thun
Interlaken Ost
Lausanne
Montreux
Genève
Genève-Aeroport
Coppet
Brig
Visp
Sion
Bellinzona
Lugano
Chiasso

Figure 1: Visualization of the long-distance traffic in Switzerland. Different lines correspond to different service-routes.



IC 1	Genève-Aéroport–Bern–Zürich HB–St.Gallen
IC 2	Zürich HB–Zug–Lugano
IC 3	Basel SBB–Zürich HB–Chur
IC 4	Zürich HB–Schaffhausen
IC 5	Genève-Aéroport/Lausanne–Biel/Bienne–Zürich HB(–St.Gallen)
IC 6	Basel SBB–Bern–Brig
IC 8	Brig–Bern–Zürich HB–Romanshorn
IC 21	Basel SBB–Luzern–Lugano
IC 51	Basel SBB–Biel/Bienne
IC 61	Basel SBB–Bern–Interlaken Ost
IR 13	Zürich HB–St.Gallen–Chur
IR 15	Genève-Aéroport–Lausanne–Bern–Luzern
IR 16	Bern–Olten–Brugg AG–Zürich HB
IR 17	Bern–Burgdorf–Olten(–Zürich HB)
IR 26	Basel SBB–Luzern–Erstfeld(–Airolo)
IR 27	Basel SBB–Olten–Luzern
IR 36	Basel SBB–Brugg AG–Zürich HB(–Zürich Flughafen)
IR 37	Basel SBB–Aarau–Zürich HB(–St.Gallen)
IR 46	Zürich HB–Zug–Erstfeld(–Airolo)
IR 70	Luzern–Zürich HB
IR 75	Luzern–Zürich HB–Konstanz
IR 90	Genève-Aéroport–Lausanne–Brig
RE	Bern–Burgdorf–Olten
RE	Erstfeld–Lugano(–Milano)
RE	Zürich HB–Chur
RE	Aarau–Zürich HB
RE	Zürich HB–Bülach–Schaffhausen
RE	Bern–Biel/Bienne
RE	Genève–Lausanne–Vevey

Figure 2: National service-routes that contribute to the long-distance traffic in Switzerland. Not included in this figure is the EuroCity (EC) and the InterCity Express (ICE), as they are international route-services.

Figure 1 shows the area that is covered by the long-distance traffic. Figure 1 already indicates the complexity of the swiss railway network. It is visible that not all stations are served equally by routes. Further, one can identify that the axis between "Zürich HB" and "Bern" is served by many different routes. It is also visible that the long-distance railway network is characterized by two major axes along North-West to South (Basel SBB – Luzern – Lugano) and West to East (Genève – Bern – Zürich HB – St. Gallen). In Figure 2 all routes are listed that contribute to the long-distance railway traffic in Switzerland. In sum there are 10 InterCity (IC) routes, 12 InterRegio (IR) routes and 7 RegioExpress (RE) routes. Not listed here are the European long-distance routes InterCity Express (ICE) and the EuroCity (EC). The ICE is operated by the german railway agency DeutscheBahn (DB) in corporation with SBB. This ICE routes connect major cities of Germany with major cities (Bern, Chur, Zürich HB, Basel and more) from Switzerland. As the SBB operates these services within Switzerland they will also be included in this study. The same applies to the EC routes.

The data used for this study is provided by the open data platform "Open Data Platform Swiss Public Transport" (ODPST)[1] operated by SBB on behalf of Switzerland's federal office of transport (BAV). On this platform SBB publishes customer information data on public transport in Switzerland. Hence, they provide access to specific public transport services that are free of charge. ODPST provides data that also includes data from other local operating agencies in Switzerland. The data sources for railway networks are mostly operational planning and controlling systems.

3.1 Long-distance traffic delays

This section should give a brief overview of delays within the long-distance traffic in Switzerland's railway network. It must be stated that Switzerland's railway services are among the most punctual in the whole world. Nevertheless, delays are still occurring and can be frustrating for all involved persons. As we can see in figure 3 the very big majority of trains arrive on-time. Around one-third of all arrivals deviate from the scheduled arrival by one minute. The small coloured in deep red represents the number of delays, which are larger than 3 minutes. 3 minutes is also the threshold used by SBB. However, SBB does not measure the delay of train-services instead

[1] https://opentransportdata.swiss/en/

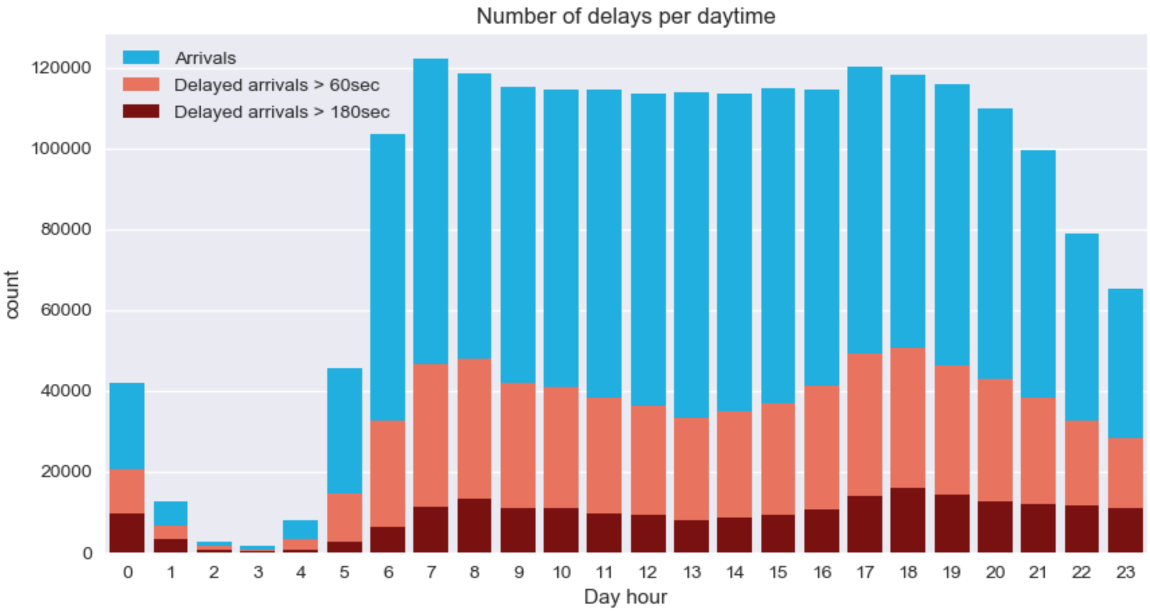


Figure 3: Delay over daytime. Delayed arrivals are highlighted in red. A delay is defined as a deviation from the actual schedule.

they measure punctuality. SBB defines punctuality from a customer’s-perspective, which means customers are delayed if they arrive at their destination with a delay larger than 3 minutes (SBB, 2018). Furthermore, in figure 3, we can see that during the morning hours between 7am and 8am there are more registered train arrivals and slightly more delays. The same can be identified between 5pm and 7pm. However, the number of delays seems growing proportionally to the growing number of arrivals. Consequently, train services are not per-se more delayed within these peak-hours. In figure 4 the delayed arrivals for different scheduled travel time classes are visualized. It emphasizes that most train-services take 6 to 8min. Further, we can also see that the number of delayed arrivals seems not to increase with increasing travel time between two stations.

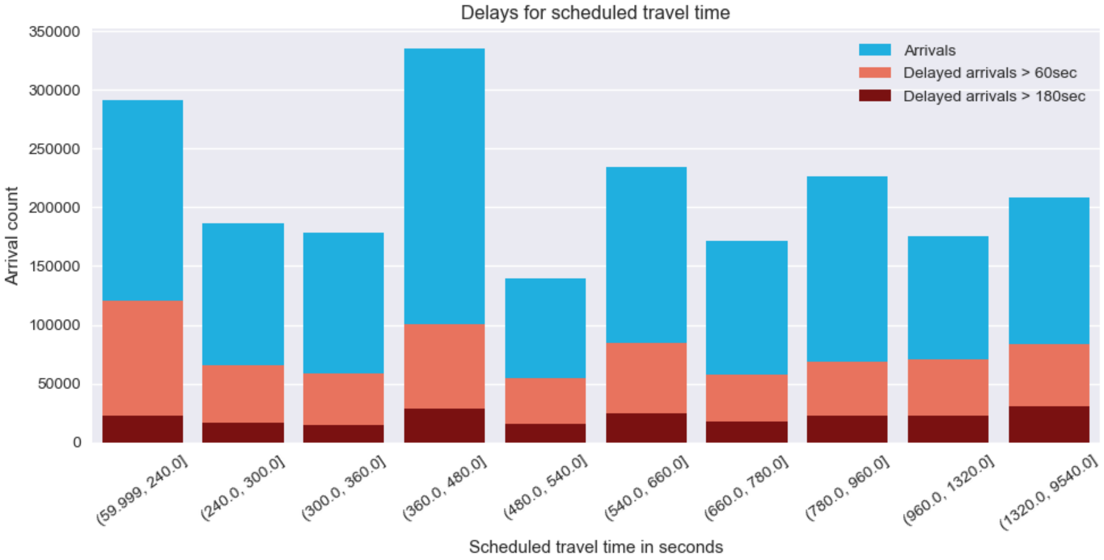


Figure 4: Number of arrivals and number of delayed arrivals grouped by scheduled travel time.

4. Methodology

In this part of the thesis, the methodology is presented that is used to predict arrival delays for trains. This part will first point out some considerations that will help to define the prediction problem and the features. Second section of the methodological part will outline how the data provided by Switzerland's public transit agencies has been pre-processed and filtered. The third section explains the methods used to extract the features from the pre-processed data that are used in the machine learning model to predict train arrival delays. But first the theoretical approach will be described. It contains defining train arrival delays, the conception of the railway network and the definition of the prediction problem. Afterwards the input feature space will be defined, which will be used by the machine learning prediction model to predict train arrival delays. An overview of the general workflow to build the prediction model is presented in the diagramm (fig. 5).

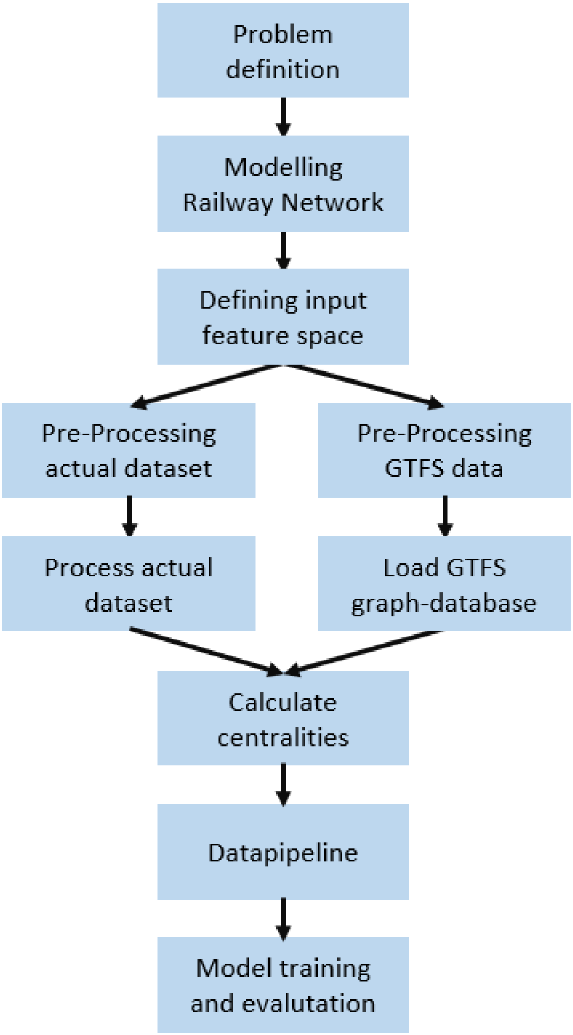


Figure 5: General workflow conducted within this thesis. For box represents a section within this chapter, indicated by the number within the box.

All the data has been pre-processed in Python programming language[2]. After first pre-processing steps the data has been treated in different environments. The GTFS data has been further loaded into the open-source graph-database Neo4j[3]. The actual time data has been solely processed in Python, where PyCharm Community[4] served as an open-source Integrated Development Environment (IDE). During processing the data in Python several different packages have been used, which will be indicated within describing the processing in detail.

4.1 Theoretical Approach

4.1.1 Defining a public transportation network

In order to approach a public transportation network in a more holistic way, a graph-based approach was proposed (see research Gap). Derrible and Kennedy (2011) emphasized in their paper "Applications of graph theory and network science to transit network design" that the field that appears particular fitted to address problems of network design is graph theory. For that reason, the following section presents how a public transportation network could be modelled within a graph-based approach. In this work the train transportation network is modelled using a time-expanded approach, similar as proposed by Huang et al. (2018). The advantage of using this approach, is that timetable information can easily visualized graphically for a better understanding (Fortin, Morency and Trépanier, 2016). In general, a railway transportation system consists of a set of unique stations $\boldsymbol{s_i} \in \boldsymbol{S}$. A train connection $\boldsymbol{c} \in \boldsymbol{C}$ is a tuple consisting of two stations $\boldsymbol{c_{i\to j} = (s_i, s_j).}$ A connection has no stops in between. A connection gets instantiated by a train departing from the connection origin-station $\boldsymbol{s_i}$ at time $\boldsymbol{dt_i}$ and arriving at the connection destination-station $\boldsymbol{s_j}$ at time $\boldsymbol{at_j}$ In this case, a connection can be interpreted as a leg of a trip $\boldsymbol{l} \ni \boldsymbol{c}$, which is defined by serving two subsequent stations $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$ at a specific time. Consequently, a service $\boldsymbol{l}$ can be defined as a list of all its sequential legs, $\boldsymbol{l = (c_{1\to 2}, c_{2\to 3}, \dots, c_{(n-1)\to n})}$, which in turn represents the whole travel of the train and can also be referred to as a rail transportation service and synonymous to a trip. A route $\boldsymbol{r}$ refers to a set of trips visiting the same stations: $\boldsymbol{r = \{l_{1,} l_2, \dots, l_n\}}$. $\boldsymbol{R}$ corresponds to all routes within the railway transportation network and therefore $\boldsymbol{r^v}$ denotes a specific route within the railway transportation network. In this thesis, $\boldsymbol{R}$ corresponds to all routes within the long-distance railway traffic (see fig 1.). In this way the graphical representation of the railway transportation network can be visualized as presented in figure 6. In table 1 the above introduced elements of the defined railway transportation network are summarized.

[2] https://www.python.org

[3] https://www.neo4j.com

[4] https://www.jetbrains.com/pycharm/

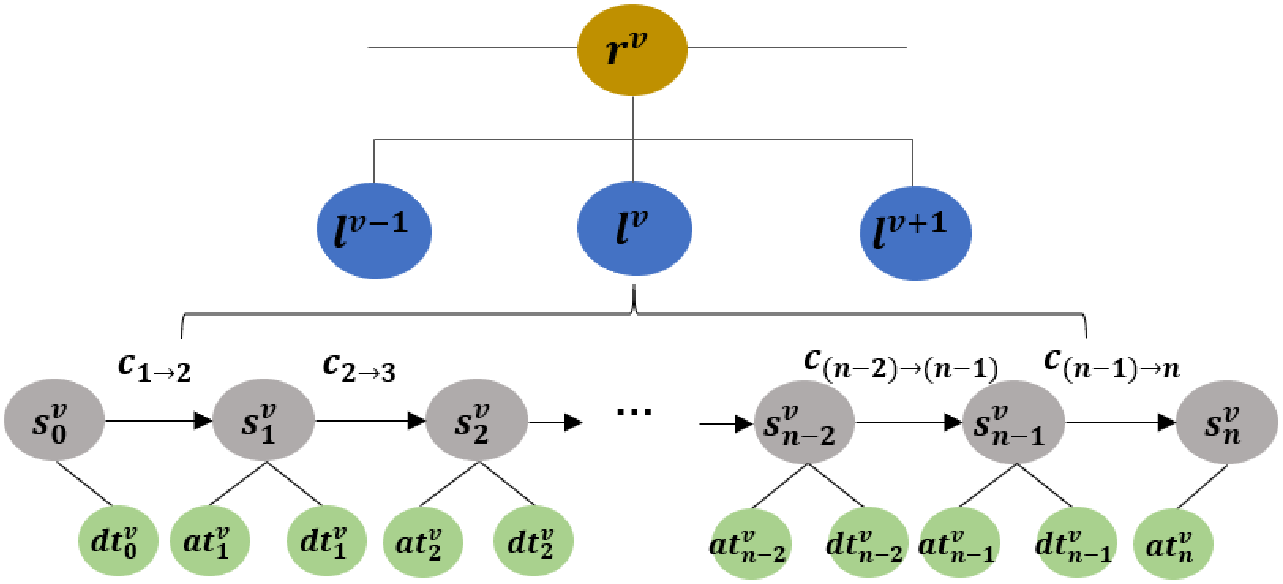


Figure 6: a graphical visualization of the timetable information of route r^v using a time-expanded modelling approach.

Element	Definition	Description
$\boldsymbol{S}$		all Stations within the Network
$\boldsymbol{R}$		all Routes within the Network
$\boldsymbol{s_i}$	$\in \boldsymbol{S}$	a station within the network
$\boldsymbol{c_{i\to j}}$	$\coloneqq (\boldsymbol{s_i}, \boldsymbol{s_j})$	connection between two subsequent stations
$\boldsymbol{r^v}$	$\in \boldsymbol{R}$	a specific route
$\boldsymbol{l^v}$	$\in \boldsymbol{r^v}$	a train-service of a route (also referred as trip)
$\boldsymbol{c_{i\to j}^v}$	$\in \boldsymbol{l^v}$	a leg of a train-service departing from station $\boldsymbol{s_i}$ and arriving at station $\boldsymbol{s_j}$
$\boldsymbol{s_i^v}$	$\in \boldsymbol{c_{i\to j}^v}$	leg origin-station of the train-service
$\boldsymbol{s_j^v}$	$\in \boldsymbol{c_{i\to j}^v}$	leg destination-station of the train-service
$\boldsymbol{dt_i^v}$	$\in \boldsymbol{s_i^v}$	Departure time of a train-service at station $\boldsymbol{s_i^v}$
$\boldsymbol{at_i^v}$	$\in \boldsymbol{s_i^v}$	Arrival time of a train-service at station $\boldsymbol{s_j^v}$

Table 1: summarizes the most important elements of a public transportation network.

4.1.2 Defining delay

A train on a trip $\boldsymbol{l^v}$ runs through all stations that are defined in its legs $\boldsymbol{c^v_{i \to j}} \in \boldsymbol{l^v}$. For reach station within $\boldsymbol{l^v}$, the train running the trip should arrive at time $\boldsymbol{at^v_i}$ and departure at time $\boldsymbol{dt^v_i}$ from station $\boldsymbol{s_i}$. These arrival and departure times are defined in a timetable for each trip and are defined during the scheduling process of the railway operating agency. Note that for $\boldsymbol{s^v_i},\ \boldsymbol{i = 1}$ there is no arrival time. Logically there exists no departure time for the end station of the trip.

Along a trip $\boldsymbol{l^v_i}$ there are different forms and possibilities a service delay can occur. For example, a trip could be defined as delayed if the vehicle arrives later than scheduled at the last station of its sequence, the trips end station respectively. A trip could also be defined as delayed, if the service departs later than scheduled of one of its stations.

An arrival delay could also be defined for any station $\boldsymbol{i}$ as the difference between the actual arrival time $\boldsymbol{\widetilde{at^v_i}}$ and the scheduled arrival time $\boldsymbol{at^v_i}$:

$$\boldsymbol{y^v_i = \widetilde{at^v_i} - at^v_i\,, \qquad y^v_i > 0} \tag{4}$$

For service punctuality overall departure delays are equally important as arrival delays. However, by consider a customer plans his journey with the aim to arrive at his destination as planned. A delayed departure of the service will not affect the customer as long as the service arrives at customer's destination as planned. Of course, a delayed departure might lead to a delayed arrival, but from a customer's perspective we could assume the main interest would be to arrive on time. Therefore, within this thesis a delay is defined as arrival time at a specific station, which is not equal to scheduled arrival time, as formulated in (4). Hence, a train that arrives at the scheduled time is considered as on-time.

4.1.3 Defining the prediction problem

In the following variables that result from a prediction are marked with a "^" on the top. Variables that result from an operation with a predicted value are also marked with a "^" because these variables do not reflect the true value, but rather an estimation.

Within this thesis "prediction problem" refers to the question, which value the prediction model predicts by considering a set of input variables in order to fulfil the prediction purpose. Regarding delay prediction, multiple approaches to solve this prediction problem exist. The purpose of delay prediction is to assess the arrival delay of a public transportation service for a specific station along its trip. To predict delays in public transit systems various approaches can be used. As outlined in the previous chapter an approach is to predict the travel time between two subsequent stations. By adding the predicted travel time to the departure time, one can extrapolate the predicted arrival time. Subsequently, the estimated delay $\boldsymbol{\hat{y}^v_j}$ for station $\boldsymbol{s^v_j}$, can be calculated as the difference between the extrapolated arrival time and scheduled arrival time. In this case the delay prediction problem can be formulated into the following two steps for the leg c^v_j , with leg destination station:

$$\boldsymbol{\widehat{at}^v_j = \hat{T}{}^v_{ij} + dt^v_i} \tag{5}$$

$$\boldsymbol{\hat{y}^v_j = \widehat{at}^v_j - at^a_j} \tag{6}$$

where $\widehat{\boldsymbol{at}}_{\boldsymbol{j}}^{\boldsymbol{v}}$ is the extrapolated arrival time at station $\boldsymbol{s}_{\boldsymbol{j}}^{\boldsymbol{v}}$, which is derived the sum of the predicted travel time $\widehat{\boldsymbol{T}}_{\boldsymbol{ij}}^{\boldsymbol{v}}$ between the stations $\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{v}}$ and $\boldsymbol{s}_{\boldsymbol{j}}^{\boldsymbol{v}}$, which form as a tuple leg $\boldsymbol{c}_{\boldsymbol{i}\to\boldsymbol{j}}^{\boldsymbol{v}}$. As outlined in the previous chapter there are different possibilities and methods to derive the prediction value $\widehat{\boldsymbol{T}}_{\boldsymbol{ij}}^{\boldsymbol{P}}$ such as machine learning methods, historical data analysis or by the snapshot principle. Travel time prediction is especially used for public bus transportation systems (Gurmu and Fan, 2014; Gal *et al.*, 2017; As and Mine, 2018; Sun *et al.*, 2018). Because this approach is particularly suited if the vehicle position is tracked over time, which is the case for AVL and APC systems in bus transportation systems. Therefore, the advantage of predicting travel time from a specific location to the next station allows transportation agencies to assess travel time from any location to the next station. Consequently, it is possible to estimate future arrival delays for the following station from any location. This advantage could especially lead to valuable input within multi-component operational prediction systems as described in chapter 2.2.4. However, for train delay prediction this approach seems to be less suitable if we consider that the most important factor causing delay in bus transportation is private traffic on the network. Travel time prediction is therefore a promising approach, if the private traffic can be modelled accordingly. But this is not the case for railway networks. For example, if there is much traffic on a road network, the individual vehicles might still be moving but slowly and close together, whereas trains do not because of track occupation restrictions, or a train gets re-scheduled and is diverted using another railway corridor with higher capacity. Therefore, travel time for a leg $c_{\boldsymbol{j}}^{\boldsymbol{v}}$ can be considered more constant in railway operations than in bus operations.

The second approach to predict delay in a transit system is characterized by predicting directly the delay $\widehat{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{v}}$ at station $\boldsymbol{s}_{\boldsymbol{j}}^{\boldsymbol{v}}$. As delays possibly occur at any station along the train trip $\boldsymbol{l}^{\boldsymbol{v}}$, the prediction problem needs to pay attention to all arrivals during a trip. The number of arrivals within a trip is equal to the number of legs $\boldsymbol{c}_{\boldsymbol{i}\to\boldsymbol{j}}^{\boldsymbol{v}} \in \boldsymbol{l}^{\boldsymbol{v}}$, which corresponds to the number of stations $\boldsymbol{m-1}$ within $\boldsymbol{l}^{\boldsymbol{v}}$. Therefore, the predictive model needs to predict delay $\widehat{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{v}}$ for:

$$\widehat{\boldsymbol{y}}_{\boldsymbol{j}}^{\boldsymbol{v}}\ \forall\ \boldsymbol{c}_{\boldsymbol{i}\to\boldsymbol{j}}^{\boldsymbol{v}}\,, \qquad \boldsymbol{c}_{\boldsymbol{i}\to\boldsymbol{j}}^{\boldsymbol{v}} \in \boldsymbol{l}^{\boldsymbol{v}} \tag{7}$$

This approach has been used in predictive analytics application in the railway transportation domain (Marković *et al.*, 2015; Ghofrani *et al.*, 2018; Oneto *et al.*, 2018). Different methods have been evaluated to predict the delay, such as stochastic event graph models (Berger *et al.*, 2011) or machine learning models as linear regression, random forests, support vector machine and artificial neural networks (Marković *et al.*, 2015; Wang and Work, 2015; Oneto *et al.*, 2016, 2018). Unfortunately, the available data within this thesis does not contain any vehicle location information. Therefore, the advantage of delay estimation using travel time prediction is not usable. As suggested by Marković et al. (2015) establishing a functional relation between train delays and various characteristics of a railway system is highly desirable. They argue that such a functional relation would allow planners to evaluate how changes in the system would affect delays, and thereby help them determine the changes that would reduce delays in the most economical way. Although, it is worth noting that the intrinsic dynamic and time varying nature of the delay phenomenon must be considered, which is mainly due to different factors (Oneto *et al.*, 2017). For this purpose, it seems more suitable to predict directly the arrival delay by using various characteristics of the railway network that could potentially influence punctuality of a train's arrival at a station. The difficulty in predicting train delays is that the punctuality of each train also depends on the punctuality of other trains within the network. Therefore, on the one hand, railway networks consist of many internal dependencies and are highly interconnected. On the other hand, trains are exposed to many other network external factors that can influence the punctuality of a train. The true underlying functional relation between train delays and various characteristics of a railway system as proposed by Marković et al. (2015) can be defined as:

$$\boldsymbol{\theta} : \boldsymbol{X} \rightarrow \boldsymbol{Y} \qquad (8)$$

The input space $\boldsymbol{X}$ includes all characteristics of a railway system that influence whether a trip is delayed or not. The output space $\boldsymbol{Y}$, consequently consists of all delays as defined in (4) that arise within the train service operations. The output space $\boldsymbol{Y}$ can be deduced by comparing the scheduled arrival times with the actual achieved arrival times, that are derived from the railway operating and controlling systems and provided by ODPST. In contrast $\boldsymbol{X}$ and $\boldsymbol{\theta}$ are unknown and therefore need to be approximated. In this thesis, $\boldsymbol{\theta}$ will be approximated by building a predictive learning-model $\boldsymbol{\vartheta}$ using the techniques and methods provided through predictive analytics. The input space $\boldsymbol{X}$ for $\boldsymbol{\vartheta}$ consists of different input variables that should capture different intrinsic railway transportation network properties, which are discussed in the next section in detail. By taking into account the prediction constraint (7) the model $\boldsymbol{\vartheta}$ needs to predict the arrival delay for each leg within a trip $\boldsymbol{l^v}$. This leads us to the problem of delay propagation within a trip $\boldsymbol{l^v}$, because an arrival delay occurring at station $\boldsymbol{s_j^v}$ will affect the arrival times of the following stations within the trip. To address this problem, the model predicts the arrival delay $\boldsymbol{\hat{y}_j^v}$ based on the train-service state at the previous station $\boldsymbol{s_i^v}$. This means, if a train-service arrived late at station $\boldsymbol{s_i^v}$, the model will take this into account for the prediction of the subsequent station $\boldsymbol{s_j^v}$. By doing so, it can be assured that delays occurring within a trip are not considered independent by the prediction model. Furthermore, the model should respect the fact that some railway network properties might change over time. In this case the prediction problem should be treated as time-series forecasting regression problem, as suggested by Oneto et al. (2016). Consequently, the model performs a regression analysis to predict delay $\boldsymbol{\hat{y}_j^v}$ at station $\boldsymbol{s_j^v}$ based on the state of the railway network at the time the train arrived at station the preceding station $\boldsymbol{s_i}$. Finally, this leads us to the definition of the prediction problem, which consists of predicting delay $\boldsymbol{\hat{y}_j^v}$ using the input features $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_m} \in \boldsymbol{X}$ at time $\boldsymbol{at_i^v}$**,** which corresponds to the arrival time of the previous station.

$$\boldsymbol{\hat{y}_j^v} = \boldsymbol{\vartheta_1}(\boldsymbol{t}) = \boldsymbol{f}\big(\boldsymbol{X}(\boldsymbol{t})\big)\,, \qquad \boldsymbol{t} = \boldsymbol{at_i^v} \qquad (9)$$

Until now, the prediction model predicts solely the arrival delay of the subsequent station. Nevertheless, this might not always satisfy the needs of railway operators, as train-dispatcher might want to know how the delay is propagated for the train service in multiple following station. Based on this requirement, a second and third model is introduced, which predicts the train arrival delay for the following two destination-stations of legs $c^{v}_{\boldsymbol{j}\rightarrow(\boldsymbol{j+1})}$ and $c^{v}_{(\boldsymbol{j+1})\rightarrow(\boldsymbol{j+2})}$ based on the state of the railway network at the same time as the first model $\boldsymbol{\vartheta_1}(\boldsymbol{t})$.

$$\boldsymbol{\hat{y}_{j+1}^v} = \boldsymbol{\vartheta_2}(\boldsymbol{t}) = \boldsymbol{f}\big(\boldsymbol{X}(\boldsymbol{t})\big)\,, \qquad \boldsymbol{t} = \boldsymbol{at_i^v} \qquad (10)$$

$$\boldsymbol{\hat{y}_{j+2}^v} = \boldsymbol{\vartheta_3}(\boldsymbol{t}) = \boldsymbol{f}\big(\boldsymbol{X}(\boldsymbol{t})\big)\,, \qquad \boldsymbol{t} = \boldsymbol{at_i^v} \qquad (11)$$

Using all three models together it allows to largen the prediction horizon of the train delay prediction system, thus making predictions that lie further in the future. The figure 7 depicts schematically how the proposed delay prediction system and its three models.

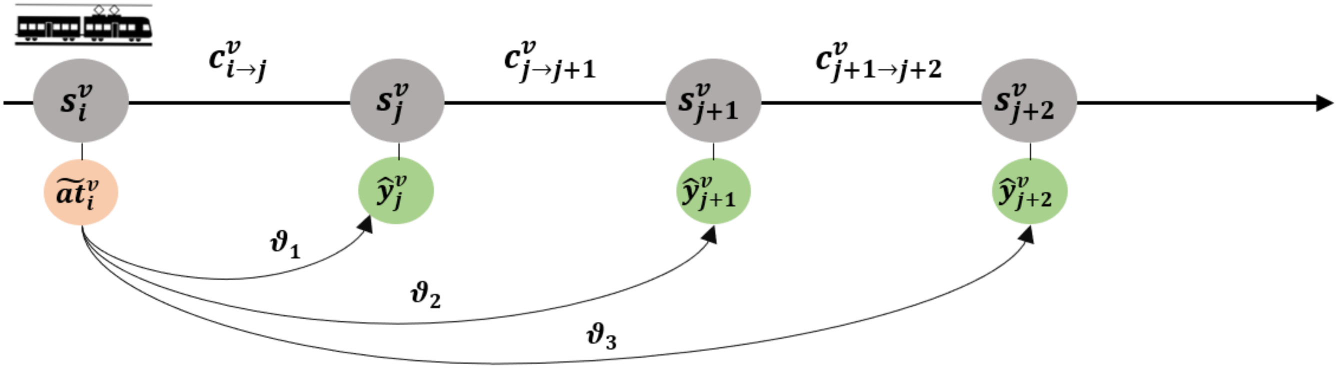


Figure 7: schematic overview of the proposed train delay predictions

For simplification reasons from now on the notation $\boldsymbol{\vartheta_1} \to \boldsymbol{\hat{y}^v_j}$ corresponds to: “the prediction of $\boldsymbol{\hat{y}^v_j}$ by $\boldsymbol{\vartheta_1}$. In the next section the input features are going to be introduced, which will be used to predict the train arrival delays. Nevertheless, as data pre-processing is a very time-consuming task in machine-learning the following assumptions have been made, to slightly simplify data pre-processing and feature engineering

4.2 Defining input feature space

As proposed the by Marković et al. (2015) the overall aim of a transportation delay prediction model, should be to identify the functional relation between train delays and various characteristics of the transportation system. For this, in the existing literature different input features have been proposed, especially trip- and network-related. In order to investigate on the formulated research questions, the input features have been categorized into five different categories, characterized by the kind of information they contribute to the prediction. In order to investigate on the formulated research questions, the input features have been categorized into five different categories, characterized by the kind of information they contribute.

As the proposed train arrival delay prediction system uses three different models ($\boldsymbol{\vartheta_1}, \boldsymbol{\vartheta_2}, \boldsymbol{\vartheta_3}$), see figure 7. Input feature spaces $\boldsymbol{X_1}$, $\boldsymbol{X_2}$, $\boldsymbol{X_3}$ correspond to the input feature space used for model $\boldsymbol{\vartheta_1}$, $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$. But in order to address, the intrinsic dynamics of a transportation network the following assumptions have been made.

> **Assumption 1:** For $\boldsymbol{\vartheta_1} \to \boldsymbol{\hat{y}^v_j}$, model $\boldsymbol{\vartheta_1}$ takes into account the characteristics of the two subsequent legs $\boldsymbol{c^v_{j\to j+1}}$ and $\boldsymbol{c^v_{j+1\to j+2}}$ within the trip $\boldsymbol{l^v}$ based on the situation at time $\boldsymbol{\widetilde{at}^v_i}$.
>
> **Assumption 2:** For $\boldsymbol{\vartheta_2} \to \boldsymbol{\hat{y}^v_{j+1}}$, model $\boldsymbol{\vartheta_2}$ takes into account the characteristics of leg $\boldsymbol{c^v_{j\to j+1}}$ and the subsequent leg $\boldsymbol{c^v_{j+1\to j+2}}$ within the trip $\boldsymbol{l^v}$ based on the situation at time $\boldsymbol{\widetilde{at}^v_i}$.
>
> **Assumption 3:** For $\boldsymbol{\vartheta_3} \to \boldsymbol{\hat{y}^v_{j+2}}$, the model $\boldsymbol{\vartheta_3}$ takes into account the characteristics of the two preceding legs $\boldsymbol{c^v_{i\to j}}$ and $\boldsymbol{c^v_{j\to j+1}}$ within the trip $\boldsymbol{l^v}$ based on the situation at time $\boldsymbol{\widetilde{at}^v_i}$.

The first assumption enables $\boldsymbol{\vartheta_1}$ to consider variables that could impact train-service punctuality that lie ahead in the trip, for example traffic congestion at subsequent stations. The second assumption enables $\boldsymbol{\vartheta_2}$ also to consider what is ahead in the trip. But in addition, by considering the characteristics of leg $\boldsymbol{c^v_{j\to j+1}}$, $\boldsymbol{\vartheta_2}$ can consider variables that could impact train-service punctuality that lie on between the train-service and station $\boldsymbol{s^v_{j+1}}$. For example, the traffic situation at station $\boldsymbol{s^v_j}$, which the train-service needs to pass to reach $\boldsymbol{s^v_{j+1}}$. The third assumption corresponds to the same idea. By considering the characteristics of the two legs that lie between the current station $\boldsymbol{s^v_i}$ and station $\boldsymbol{s^v_{j+2}}$ might lead to better prediction accuracy.

These assumptions allow to generate one table containing all features, and after filtering each row can be used by $\boldsymbol{\vartheta_1}$, $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$ for training, evaluation and testing of the models. On the other side, assumption 1 and 2 cause the following special case:

> **Special Case:** Per definition $\boldsymbol{l^v}$ is consisting of $\boldsymbol{n}$ legs. For $\boldsymbol{\vartheta_1} \to \boldsymbol{\hat{y}^v_n}$, the characteristics of the two subsequent legs $\boldsymbol{c^v_{n\to n+1}}$ and $\boldsymbol{c^v_{n+1\to n+2}}$ do not exist and can have no impact on the punctuality of the train service arriving at $\boldsymbol{s^v_n}$. Consequently, the corresponding features are set to null. The same applies to the scenario $\boldsymbol{\vartheta_2} \to \boldsymbol{\hat{y}^v_n}$ with $\boldsymbol{c^v_{n+1\to n+2}}$.

In table 2, all input features are summarized and explained. Feature highlighted with (*) are features that exist three times corresponding to trip legs $\boldsymbol{c^v_{i\to j}}$, $\boldsymbol{c^v_{j\to j+1}}$, $\boldsymbol{c^v_{j\to j+2}}$. In table 3 the abbreviations for the corresponding feature-category are explained.

#	Feature	Definition	Datatype	Description	Category
1	date		Metric	Date of service when trip starts	TR
2	weekday		Nominal	Weekday of trip	TR
3	holiday	national holiday	Boolean	True if national holiday	TR
4	start-station	$\boldsymbol{s_0} \in \boldsymbol{l^v}$	Nominal	Trip start station	TR
5	end-station	$\boldsymbol{s_n} \in \boldsymbol{l^v}$	Nominal	Trip end station	TR
6	route-id	$\boldsymbol{r^v}$	Nominal	Specifies the route	TR
7	current station	$\boldsymbol{s^v_i}$	Nominal	Station where train-service last arrived	TR
8*	leg destination	$\boldsymbol{s^v_i}$	Nominal	leg destination station	TR
9	scheduled depart time	$\boldsymbol{dt^v_i}$	Metric	Departure time at current $\boldsymbol{s^v_i}$	TR
10*	scheduled arrival time	$\boldsymbol{at^v_i}$	Metric	Arrival time at current $\boldsymbol{s^v_i}$	TR
11*	scheduled travel time	$\boldsymbol{dt^v_j - dt^v_{0i}}$	Metric	Travel time between $\boldsymbol{s^v_i}$ and $\boldsymbol{s^v_j}$	TR

12*	station number	$\boldsymbol{i} \in \{\mathbf{1}, \mathbf{2}, \dots, \boldsymbol{n}\}$	Metric	corresponds to the station index number within $\boldsymbol{l^v}$	TR
13	trip start time	$\boldsymbol{dt_0^v}$	Metric	Departure time at $\boldsymbol{s_0^v}$	TR
14	time since start	$\boldsymbol{at_i^v - dt_0^v}$	Metric	Time since trip started	TR
15	current arrival delay	$\boldsymbol{y_i^v}$	Metric	Arrival delay at current station $\boldsymbol{s_i^v}$	DP
16	Previous arrival delay	$\boldsymbol{y_{i-1}^v}$	Metric	Arrival delay at station before current station $\boldsymbol{s_{i-1}^v}$	DP
17	pre-previous arrival delay	$\boldsymbol{y_{i-2}^v}$	Metric	Arrival delay at second station before current station $\boldsymbol{s_{i-2}^v}$	DP
18*	delays at station	$\boldsymbol{\frac{1}{m}\sum_{i=1}^{3h} y_j^i,}$	Metric	Average arrival delay at station $\boldsymbol{s_j^v}$ during last three hours	NR
19*	busy-index	$\boldsymbol{\sum_{x=1}^{t} c_{i\to j}^x,}$	Metric	Number of train-services running from $\boldsymbol{s_i}$ to $\boldsymbol{s_j}$ within $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	NR
20*	snapshot delay previous trip	$\boldsymbol{y_j^{v-1}}$	Metric	Arrival delay of previous trip for $\boldsymbol{l_{(j-1)j}}$ by same route	S
21*	snapshot delay last service	$\boldsymbol{y_j^x}$	Metric	Delay of last train-service running from $\boldsymbol{s_i}$ to $\boldsymbol{s_j}$	S
22	betweenness centrality origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
23*	betweenness centrality destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
24	indegree centrality origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
25*	indegree centrality destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
26	outdegree centrality origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
27*	outdegree centrality destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
28	closeness centrality origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
29*	closeness centrality destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR
30*	edge betweenness	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t <\ dt_i^v + 1.5h]}$	SR

31	pagerank origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t < dt_i^v + 1.5h]}$	SR
32*	pagerank destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t < dt_i^v + 1.5h]}$	SR
33	load centrality origin	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t < dt_i^v + 1.5h]}$	SR
34*	load centrality destination	see 5.3.2	Metric	Calculated for $\boldsymbol{s_i^v}$ for timeframe $\boldsymbol{[dt_i^v - 1.5h < t < dt_i^v + 1.5h]}$	SR
35	actual arrival delay at $\boldsymbol{s_j^v}$			$\boldsymbol{y_j^v}$, used only for model training, testing and validation	
36	actual arrival delay at $\boldsymbol{s_{j+1}^v}$			$\boldsymbol{y_{j+1}^v}$, used only for model training, testing and validation	
37	actual arrival delay at $\boldsymbol{s_{j+2}^v}$			$\boldsymbol{y_{j+2}^v}$, used only for model training, testing and validation	

Table 2: Summarizes all input features used to predict delay $\hat{y}_j$ at station s_j for time $t = t_{j-1}^a$. The last three rows correspond to the actual arrival delays. They are needed to train, test and validate the machine-learning model

As defined in the prediction problem (9) all the features defined in table 2 have to be calculated or extracted for $\boldsymbol{t = at_i^v}$ in order to predict delays $\boldsymbol{\hat{y}_j^v}$, $\boldsymbol{\hat{y}_{j+1}^v}$, $\boldsymbol{\hat{y}_{j+2}^v}$ at station $\boldsymbol{s_j^v}$, $\boldsymbol{s_{j+1}^v}$, $\boldsymbol{s_{j+2}^v}$. In the table below the input feature categories are listed. In the following section these categories will be explained more in detail.

Category	**Abbreviation**
Trip-related	TR
Delay propagation	DP
Snapshot	S
Station-related	SR
Network-related	NR

Table 3: Input feature categories and their abbreviation.

Table 2 corresponds to the table schema of the input table that will be used to train, validate and test the prediction model. Each record in the input table, corresponds to one observation for which the arrival delay needs to be predicted using the variables in table 2

4.2.1 Trip-related features (TR)

The input features (1) to (13) are input features that characterize specific elements within a trip. Most of these features are commonly used as input for predictive models (Yaghini, Khoshraftar and Seyedabadi, 2013; Marković *et al.*, 2015; Oneto *et al.*, 2016; Zychowski, Junosza-Szaniawski and Kosicki, 2018). These features capture mostly categorical information, such as the route (6) and its start- (4) and end-station (5). But as the model predicts based on $\boldsymbol{t = at_i^v}$, the current station (7) and leg destination station (8) can also be extracted from the timetable. The current station is defined as the station, where the train service last arrived at $\boldsymbol{t = at_i^v}$. These features are used as input based on the assumption that delays may arise depending on the destination station. For example, a highly frequented station might register more arrival delays as a station with only few arrivals a day. Using the service's date (1) one can identify the weekday (2) and check whether it was a national holiday (3) or not. Scheduled departure (9) and arrival (10) time correspond to the times fixed in the timetable. These features (1), (2), (3), (9) and (10) should capture the daily and weekly distribution of arrival delays. The input feature "station number" (11) corresponds to the index number $\boldsymbol{n}$ within the the trip $\boldsymbol{l^v}$of the leg destination station. The indices start with value 0, therefore the station number of the start-station (4) is always equal to 0, as it is the first station of the ordered sequence that characterizes $\boldsymbol{l^v}$. Input feature (12) corresponds to the departure of the trip at the first station of his trip, whereas (13) denotes how long the trip is already during.

4.2.2 Delay propagation features (DP)

In this thesis delay propagation relates to the current delay of the trip $\boldsymbol{l^v}$. The assumption is that if the train-service arrived delayed station $\boldsymbol{s_i^v}$ it is likely that this delay will be propagated over all following legs $\{\boldsymbol{c_{i\to j}}, .., \boldsymbol{c_{(n-1)\to n}}\} \in \boldsymbol{l^v}$. In order to account this, the delay propagation feature (14-16) are considered for predicting the delays $\boldsymbol{\hat{y}_j^v}$, $\boldsymbol{\hat{y}_{j+1}^v}$ and $\boldsymbol{\hat{y}_{j+2}^v}$. The input features capture the train-service delay for the current station and the two previous stations. As a train-service might be able to catch up small delays during a leg, the last three arrival delays are considered. Therefore, it should be able to distinguish between a train-service that is constantly delayed over all stations or a small delay, that can be catched up as we can assume delayed trains run faster to reduce their delay (Goverde, 2010). In the proposed train delay prediction system, at the start of a trip at station $\boldsymbol{s_0^v}$ the delay propagation features (14-16) are set to 0 until a corresponding delay has been registered. An exception for the lines "ICE" and "EC" have been put in place. These two lines are incoming from abroad (Germany and Italy), therefore $\boldsymbol{s_0^v}$ corresponds to the first registered arrival station in Switzerland. At this point these train-services might already be delayed. In this case the delay propagation feature (14) is set to the arrival delay of the first arrival station in Switzerland. Feature 15 and 16 are set to 0 because, this data is not available because it was registered outside of Switzerland.

4.2.3 Snapshot features (S)

Snapshot features are features that follow the snapshot principle proposed by Senderovich et al. (2014) and applied to transportation network as proposed by Gal et al. (2017) and Sun et al. (2018) as discussed in chapter 2.2.3. In the outlined train delay prediction system, the snapshot feature (19) refers to the registered delay $\boldsymbol{y_j^{v-1}}$ for $\boldsymbol{c_{i\to j}^{v-1}}$ of the train-service $\boldsymbol{l^{v-1}}$, which was executed before $\boldsymbol{l^v}$. The input feature (20) corresponds to the arrival delay $\boldsymbol{y_j^x}$ for $\boldsymbol{c_{i\to j}^x}$ of a train-service $\boldsymbol{l^x} \notin \boldsymbol{r^v}$ with $\boldsymbol{at_i^x} < \boldsymbol{at_i^v}$, as proposed by Gal et al. (2017) and Sun et al. (2018). It corresponds to the arrival delay of a train-service associated to another route but shares the same leg with $\boldsymbol{l^v}$.

4.2.4 Station-related features (SR)

The category station-related features capture topological properties a station in relation to the railway network. As emphasized in the research gap, public transportation systems contain the nature of physical networks as they include the combination of lines and nodes that intersect with each other (Derrible and Kennedy, 2011). It has been shown that centrality measures are able to capture different topological characteristics of railway networks and can be useful to detect and characterize important stations within a network (Derrible, 2012; Zhang *et al.*, 2013; To, 2015). On the other hand, Lee et al. (2016) stated that among the various delay factors, timetable is the most economic control factor, and the quality of a timetable is related to the punctuality of a railway system. The here proposed station-related features combine these two insights. For this, a graph-network is built based on the timetable data, where stations are represented as nodes and train-services as edges connecting the nodes. In this way, the time-dependent topological relations between the stations and the topological property of a station within the whole timetable-network can be calculated using centrality measures.

In order to account the intrinsic dynamic and interdependencies of a railway network these station-related features are calculated for the period $\boldsymbol{[dt_i^v - 1.5h < t < \ dt_i^v + 1.5h]}$ for each observation. Consequently, the centrality measure of a leg origin- and destination-station reflects the topological characteristics of the station within its timetable network during this period. In order of computational limits this time-period has been set to 1.5 hours before and after the train-service's departure time $\boldsymbol{dt_i^v}$ for leg $\boldsymbol{c_{i \to j}}$. Further the time-period has been chosen to form a time-interval around $\boldsymbol{dt_i^v}$ because it can be argued that if the timetable quality is low before $\boldsymbol{dt_i^v}$ that would still have an impact on the train-service as a railway networks is assumed to recover from delays only slowly. Therefore, centrality measures 21-33 listed in table 2, assess different topological characteristics of a station within the timetable.

In this section the centrality measures and their purpose and relevance in a transportation network context is discussed. Concerning calculation of the centrality measures a separate section is dedicated 5.3.2.

- Degree centrality (23, 24, 25, 26)

Degree centrality of a node is a local metric that refers to the number of edges that are connected to a node (Psaltoglou and Calle, 2018). It therefore captures the topological properties of a station according to the stations it is connected by a train-service. Consequently, the proposed features indegree-centrality and outdegree-centrality refer to the number of ingoing and outgoing train-services at the leg origin- and the leg destination-station within $\boldsymbol{[dt_i^v - 1.5h < t < \ dt_i^v + 1.5h]}$.

- Betweenness centrality (21,22)

Contrary, to degree centrality, betweenness centrality of a node is a global metric, as it considers the whole network for its calculation. Therefore, it captures the topological property of a station within the whole timetable-network. Betweenness centrality is based on pair-wise shortest path connections between all node pairs in a network. The betweenness centrality of a node corresponds to the number of shortest paths passing the node. Therefore, nodes with high a betweenness centrality correspond to nodes that lie between many other nodes (Psaltoglou and Calle, 2018). Nodes with a betweenness centrality are often emphasized as network critical nodes (To, 2015). In transportation networks betweenness centrality highlights the importance of a station as a transfer point between any pairs of nodes (Derrible, 2012).

- Edge betweenness centrality (29)

The edge betweenness centrality equivalent to betweenness centrality of a node. It is defined as the number of the shortest paths that go through an edge in a network (Lu and Zhang, 2013). Therefore, an edge with a high edge betweenness centrality can be interpreted as a bottleneck edge or a bridge between two sub-groups in network (Pandey and Kemper, 2016).

- Closeness centrality (27, 28)

As betweenness centrality, closeness centrality is also a global metric. Instead of considering the number of shortest path going through a node, a node's closeness centrality is determined by average length of the shortest path between the node and all other nods (Freeman, 1979). Therefore, a node with a high closeness centrality can be easily reached by any other node. In a railway network a station with low closeness centrality is a node that is badly connected to all other stations.

- PageRank (30, 31)

Is a centrality measure that is based on PageRank algorithm, which computes the importance of nodes based on the number of incoming links, the link propensity of the linkers and the centrality of the linkers. In transportation network PageRank indicates transportation hubs of multiple stations, that are well connected within each other (Huang *et al.*, 2018).

- Load centrality (32, 33)

Load centrality is very close related to betweenness centrality. As betweenness centrality its calculation is based on pair-wise shortest path connections between all node pairs in a network. Load centrality was introduced to capture the collaborative ties between each node-pair (Newman, 2001).

4.2.5 Network-related features (NR)

Are features as introduced by Oneto et al. (2016) and Wang and Work (2015). The basic idea behind, this kind of feature is to capture the current railway traffic situation on the whole network. As Wang and Work (2015) concluded that taking into account other delayed train-services in the network does not improve the prediction accuracy. They argued that this can be explained because once a train is delayed at a station, it is observed that the delay will propagate for several stations and therefore is already considered in the delay propagation feature. But another explanation could be that solely taking into account train-services departing from neighboring station of the leg origin-station is not sufficient. Therefore, within this thesis two new network-related features are proposed.

- Delays at station (18)

The input feature "Delays at station" corresponds to the average arrival delay of all incoming train-services within $\boldsymbol{at_i^v - 3h}$ at the leg destination-station. For this, the delay of all incoming train-services at the leg destination-station have been summed and averaged by the number of train-services. Using, this approach also non-neighboring stations from the leg-origin station are considered. Furthermore, a station registering a high amount arrival delays within a time-period needs constantly to re-schedule trains. For example, changing the arrival-platform of a train-service. This can lead to shortages and cause delayed arrivals for train-services that have not been delayed until this point

- Busy-index (18)

The input feature "Busy-index" is a simple count function, that counts the train-services within the leg origin- and destination-station within $\boldsymbol{[dt_i^v - 1.5h < t < \ dt_i^v + 1.5h]}$ for a train-service leg $\boldsymbol{c_{i\to j}}$. Same as the station-related features, the busy-index is based on timetable data. But contrary to those the busy-index assesses the absolute train-service frequency between two stations. The term busy therefore denotes, how often the connection between two subsequent stations have been serviced, consequently a connection could refer to as busy if this connection has been serviced many times within the timeframe.

5. Data and Pre-processing

5.1 Data overview

This section outlines the data used to predict arrival delays in detail. Furthermore, it sketches how the data has been filtered and pre-processed in order to use it to train the machine learning model and to evaluate its results. As mentioned, the data used in this thesis is derived from the following open data platform.

- Open Transport Data Platform Swiss Public Transport (ODPST)
  https://opentransportdata.swiss/en/

The following table gives an overview over the specific datasets that are used in this thesis. The last column describes for which purpose the data will be used within this thesis. For this, the term “network-related information” refers to the purpose that from this dataset specific network-related information has been extracted. This can be in form of calculated metrics or categorical values. The term “trip-related information” refers to trip-specific information. This basically includes the actual arrival and departure time that was achieved by the train.

Dataset Name	Timeframe	Description	Purpose
Static GTFS	Timetable 2018	Static GTFS Feed	Timetable related information
Actual data	01.01.2018 – 30.6.2018	Train describer data containing actual arrival and departure time of each trip	Trip and network related information

Table 4: Overview of the used datasets

In general, these datasets are the basis to engineer the features, which should reflect the public transportation network in a more holistic way using a graph-based approach. They finally should contribute to answer the research question, how graph-based network properties contribute to delay prediction in public transportation. The following section will discuss how the actual data dataset has been filtered and processed in order to create the basis for the input data for the prediction model to predict arrival delays for the long-distance traffic of Switzerland’s public railway transportation network. In the second part the filtering and processing of the GTFS data will be discussed in detail. In addition, the input feature deducted from the GTFS data will be presented and discussed and put into relation to the network-based approach for delay prediction.

5.2 Actual time dataset pre-processing

5.2.1 Processing

The actual time dataset[5] contains all trips, which have been executed in reality. The corresponding documentation[6] explains in detail the table fields and their meaning. According to the documentation the dataset contains the achieved, respectively the real arrival times of past trips. The used actual time dataset is very close to what Ghofrani et al. (2018) emphasized as train describer data. According to ODPST the actual data is not available for all vehicles as it is depending whether the required information systems are available for the train. For that they use the last known forecast is used as the actual time. However, even if this are approximations the ODPST confirms that these values are still very interesting as they can be used to produce statistical evaluations about punctuality, regularity or connection quality. According to (ODPST) the data is uploaded on a daily basis for the previous day, which means the data is delayed by 24 hours. The dataset is published as comma separated values (csv) files. For each month the ODPST aggregates the corresponding .CSV files into a zipped folder and adds it to the archive on Google Drive[7]. For that reason, each zip-Folder contains between 28 to 31 .CSV files for each day of the month. After unzipping the folder each month, the folder-size is around 6 GB large. At the time starting this thesis the following time-period 01.01.2018-31.11.2018 was available. From this time period a subset from the 01.01.2018 to the 30.6.2018 (ca. 36 GB) has been downloaded and stored, which contains around 180 .CSV files and form together the actual time dataset.

In table 5 the table-schema of the dataset is presented. As you can see the dataset contains 21 fields. To identify a whole trip with all its stops, which then corresponds to the defined $\boldsymbol{l^v}$, all records with the same FAHRT_BEZEICHNER need to be selected. According to the dataset's documentation the field FAHRT_BEZEICHNER is a composition of the cells BETREIBER_ID and LINIEN_ID, to which some extended reference is added. As the ODPST publishes the actual data every day for the previous day the FAHRT_BEZEICHNER is not unique when aggregating data of several days. Nevertheless, the field FAHRT_BEZEICHNER in combination with BETRIEBSTAG can be used as unique identifiers of a trip.

In order to enable developers to assess the data quality of the actual time data records (AN_PROGNOSE and AB_PROGNOSE), are categorized into 5 different classes:

1) UNKNOWN
2) Empty (=FORECAST)
3) FORECAST
4) ESTIMATED
5) REAL

The documentation to the actual datasets states that the term "UNKNOWN" indicates that no forecast and actual time are available for this and all previous stops of the trip. If the cell is blank (=Empty) the actual arrival and departure times have been forecasted, same as if the record has been labelled "FORECAST". According to ODPST these records receive the actual times based on the forecast times supplied by SBB's operational planning system if the stops are part of SBB's railway network infrastructure or of the railway network infrastructure of BLS AG and the Schweizerische Südostbahn AG (SOB). The documentation states that inaccuracies are possible in this

[5] Accessed by: https://opentransportdata.swiss/de/dataset/istdaten

[6] https://opentransportdata.swiss/en/cookbook/actual-data/

[7] https://drive.google.com/drive/folders/1SVa68nJJRL3qgRSPKcXY7KuPN9MuHVhJ

situation and are insignificant. Records are labelled with “ESTIMATED” if the actual arrival times are derived by the control system of the railway infrastructure. This control system captures the time for specific train before entering the station. The actual arrival time is then estimated based on this time capture. The term “REAL” is assigned if the effective arrival time is available.

Field Name	Datatype	Description
BETRIEBSTAG	DD.MM.YYYY	The day the trip took place
FAHRT_BEZEICHNER	String	Corresponds to a trip-id (unique when combined with BETRIEBSTAG)
BETREIBER_ID	String	Business organization number of the operating agency
BETREIBER_ABK	String	abbreviated operating agency name
BETREIBER_NAME	String	Spelled operating agency name
PRODUKT_ID	String	Ship, Bus, Train or etc. → Transportation mean
LINIEN_ID	Integer	A purely technical key for each trip (is unique on a daily basis)
LINIEN_TEXT	String	Corresponds to a route as in figure 2 (section Data availability)
UMLAUF_ID	String	Vehicle number and the time it is operating
VERKEHRSMITTEL_TEXT	String	Textual description of the transportation form
ZUSATZFAHRT_TF	Boolean	True if it is an additional trip (added to schedule)
FALLT_AUS_TF	Boolean	True if the trip failed
BPUIC	Integer	Stop-ID
HALTESTELLEN_NAME	String	Spelled name of the stop
ANKUNFTSZEIT	DD.MM.YYYY HH24:MI	Scheduled arrival time
AN_PROGNOSE	DD.MM.YYYY HH24:MI:SS	Actual arrival time
AN_PROGNOSE_STATUS	Categorical	Classifies AN_PROGNOSE quality
ABFAHRTSZEIT	DD.MM.YYYY HH24:MI	Scheduled departure time
AB_PROGNOSE	DD.MM.YYYY HH24:MI:SS	Actual departure time
AB_PROGNOSE_STATIS	Categorical	Classifies AB_PROGNOSE quality
DURCHFAHRT_TH	Boolean	True if mean of transport did not stop at scheduled stop

Table 5: Schema of the raw actual dataset

Pre-processing of actual time data

The actual dataset contains also records from all operating transportation agencies in Switzerland. Therefore, the datasets include trips that are not part of the long-distance traffic, such as bus and tram trips operated by local agencies in cities. As the long-distance traffic is operated by the SBB the actual data will be filtered and processed as sketched in figure 8. The dataset is first filtered by the operating agency (step A) using the field BETREIBER_ID. Afterwards all values are dropped that were labelled “UNBEKANNT” in AN_PROGNOSE or AB_PROGNOSE. This is the case for very few records and can therefore be neglected. In step C all records are filtered, whose LINIEN_TEXT field is equally to one the national service-routes presented in figure 2 in section “Study Area and Data availability”. After completing step C, the actual contains now only the records that are related to the long-distance traffic. This pre-processing flow has been executed for each .CSV file individually, which means they have not been aggregated. This allowed to parallelize the pre-processing computation over several computation cores, where each core executed the sketched workflow 8 for a whole .CSV file. The pre-processed CSV file contains about 12’000 records for each day, which corresponds to around 1620 trips a day. Based on the pre-processed actual data, the input features 1-20 as defined in table 2 have been computed in Python 3.7. This process is summarized in the following section.

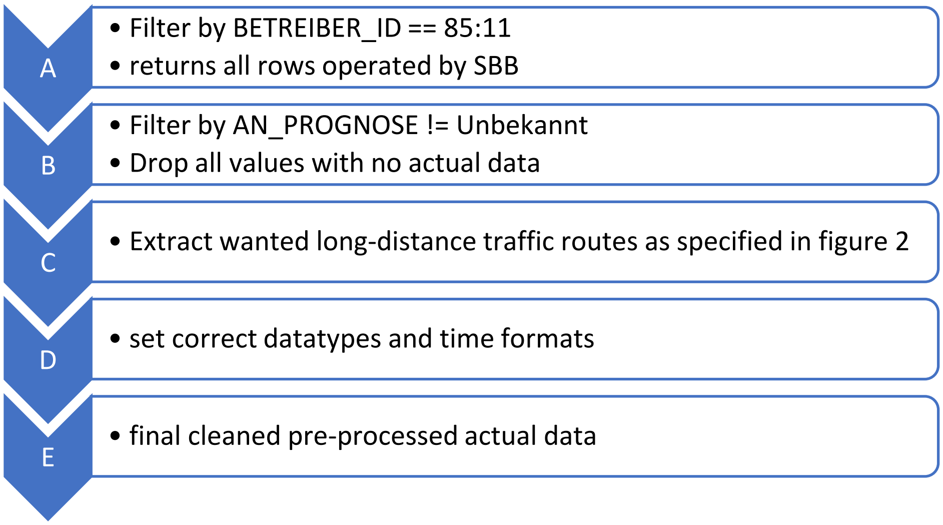
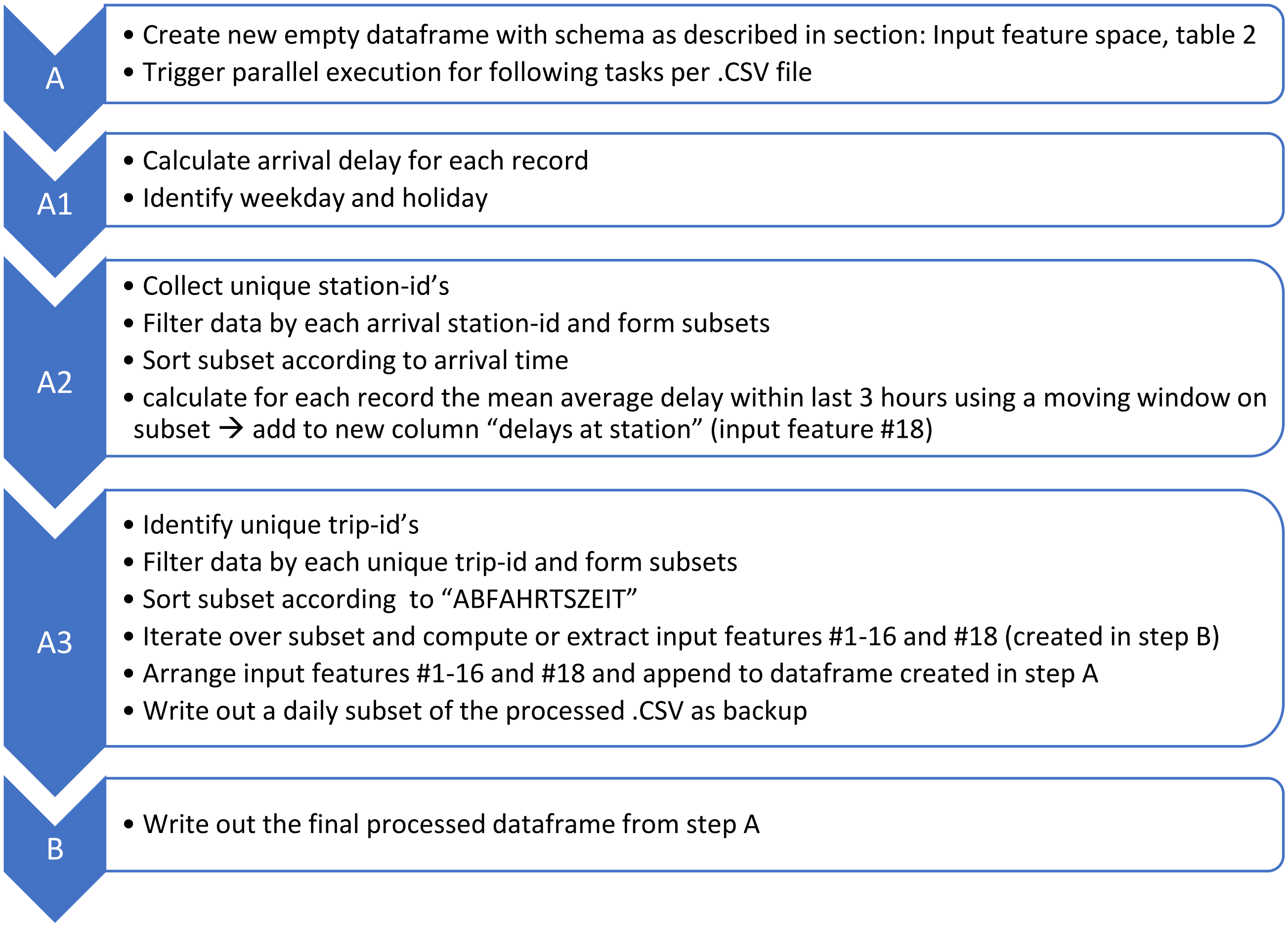

A
• Filter by BETREIBER_ID == 85:11
• returns all rows operated by SBB
B
• Filter by AN_PROGNOSE != Unbekannt
• Drop all values with no actual data
C
• Extract wanted long-distance traffic routes as specified in figure 2
D
• set correct datatypes and time formats
E
• final cleaned pre-processed actual data

Figure 8: Workflow for pre-processing actual time dataset.

5.2.2 Feature engineering actual time data

After cleaning and filtering the actual time dataset it is possible to calculate the input features 1-16 and 19 as defined in table 2. In addition. the three variables $\boldsymbol{y}_{\boldsymbol{j}}^{\boldsymbol{v}}$, $\boldsymbol{y}_{\boldsymbol{j+1}}^{\boldsymbol{v}}$, $\boldsymbol{y}_{\boldsymbol{j+2}}^{\boldsymbol{v}}$, which the model should predict are also calculated within this process. For this the following workflow has been implemented in a Python 3.7 script. As sketched in figure 9, the subtasks A1-A3 form together a task that is executed on each .CSV file. Task A creates an empty table with the schema of table 2 defined section 4.2. For each .CSV file within the actual time dataset the subtasks A1-A3 are executed in parallel to speed up processing. Task A triggers the subtasks for each .CSV file according to the availability of a computational core. Further task A collects the resulting dataframe in subtask A3 to generate a one dataframe that aggregates all processed .CSV files into one .CSV file, which is written in step B. At this point all input features 1-16 and 19 have been computed. The parallel processing of the .CSV has limited the calculation of the snapshot feature 17 and 18. For train-services early in the morning the corresponding snapshot train-services can only be identified if all .CSV files are aggregated. Therefore, the snapshot features have been computed after step B in figure 9 using the resulting dataset containing all .CSV files. For the calculation of the feature "delays at station" (19) it is assumed the railway network is set back in its initial position after one night as the railway operations are not executed during the whole night.



A

- Create new empty dataframe with schema as described in section: Input feature space, table 2
- Trigger parallel execution for following tasks per .CSV file

A1

- Calculate arrival delay for each record
- Identify weekday and holiday

A2

- Collect unique station-id's
- Filter data by each arrival station-id and form subsets
- Sort subset according to arrival time
- calculate for each record the mean average delay within last 3 hours using a moving window on subset → add to new column "delays at station" (input feature #18)

A3

- Identify unique trip-id's
- Filter data by each unique trip-id and form subsets
- Sort subset according to "ABFAHRTSZEIT"
- Iterate over subset and compute or extract input features #1-16 and #18 (created in step B)
- Arrange input features #1-16 and #18 and append to dataframe created in step A
- Write out a daily subset of the processed .CSV as backup

B

- Write out the final processed dataframe from step A

Figure 9: Workflow for the calculation of input feature 1-16 and 19 as defined in table 2.

The resulting table widely corresponds to the input table that will be used for the prediction-model. Until here the input features 1-16 and 19 have been calculated as defined in table 2. In addition, the variables of interest, the one to be predicted are also calculated in this step. The actual arrival delays have been calculated as defined in section 4.1.2.

5.3 Static GTFS dataset and pre-processing

The static GTFS (General Transit Feed Specification[8]) provided by ODPST contains the timetable information for Switzerland's public transportation network. In general, GTFS describes a digital exchange format of transit data. A series of files that is arranged as specified in GTFS is called a GTFS feed or simply a GTFS dataset (Google, 2015). It has been developed by Google for timetables used for public passenger transport services and relevant geographical information (Google, 2015). GTFS facilitates data sharing and access to information for user for transit agency operational data. Due to its simplicity, small transit agencies as well as lager ones can publish their data at low cost (Fortin, Morency and Trépanier, 2016). Static GTFS contains up to 12 different comma-separated value files that are related to each other by relational keys. All these files together would represent a complete static GTFS file, which is usually aggregated and provided by a .ZIP file. GTFS data are mostly used in online applications to provide route and schedule information to transit users (Fortin, Morency and Trépanier, 2016). In research GTFS data has also been used for accessibility analysis (Fayyaz S., Liu and Zhang, 2017; Kujala *et al.*, 2018) or spatiotemporal analysis and failure detection in public transport systems (Hadas *et al.*, 2014).

The static GTFS data provided by ODPST represent the corresponding timetable of 2018, which corresponds to the actual dataset. The GTFS data provided contains 8 .CSV files that are related to each other as illustrated in figure 10.

[8] https://opentransportdata.swiss/de/cookbook/gtfs/

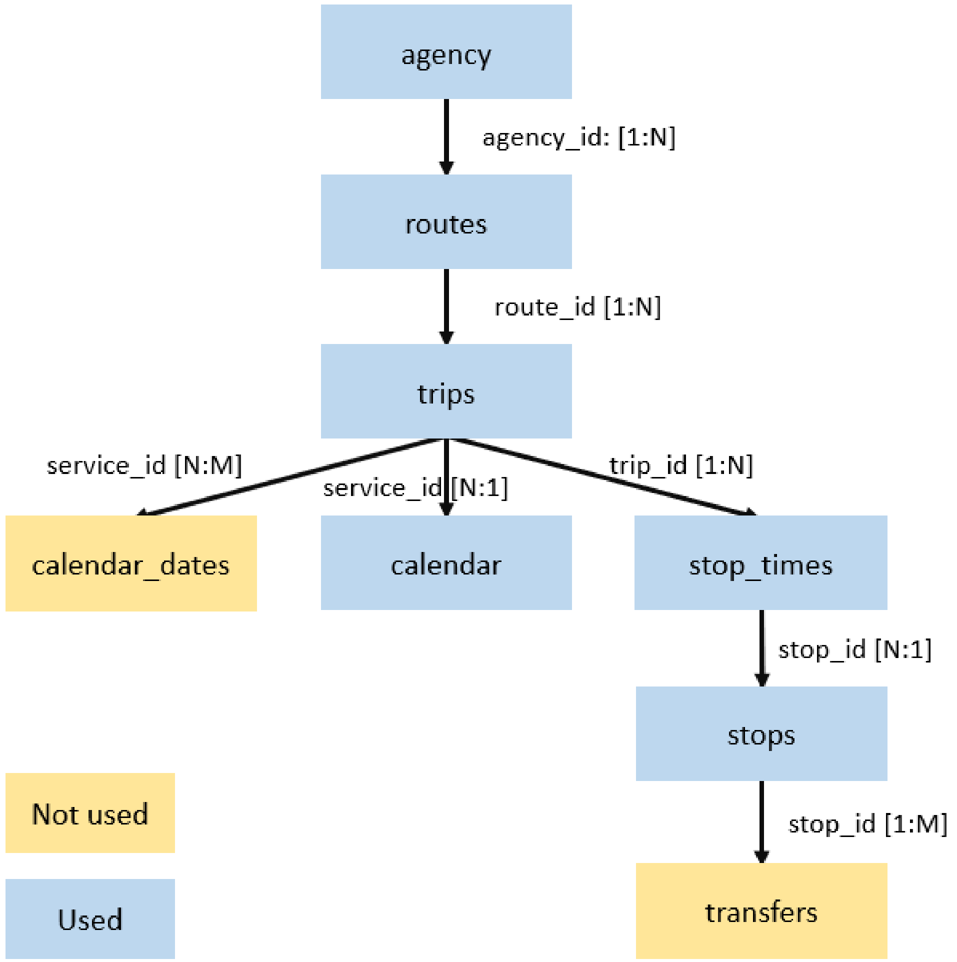


Figure 10: Illustration of the used static GTFS data and how they are related to each other. The corresponding foreign keys are indicated next to the arrows. [1: N] denotes that the foreign key matches N records in the related file. Blue coloured files have been used

As illustrated in figure 10, the railway agency with specified *agency_id* (for SBB: 11) operates multiple routes, which can be differentiated by their unique *route_id* value in the routes-file. For each route, multiple trips exist in the trips-file. Every trip is associated with multiple records in the stop_times-file, which defines the scheduled arrival and departure time of the trip at a station. Therefore, all records with the same *trip_id* value in the stop-times-file form an entire train-service $\boldsymbol{l^v}$ as defined in table 2. The stop_times reference to a specific stop, which can be identified in the stops-file using the key *stop_id*. The calendar-file contains information about the time-period within the trip is executed regularly. As the provided GTFS dataset ODPST is the valid timetable for nearly the whole year of 2018 the time-period value in the calendar-file is equal. Further the calendar-file specifies the weekday of the trip-execution, as some trips might only be executed on weekends, However, it does not specify dates for a trip-execution. The files calendar-dates and transfer have not been used within this thesis because they would not have added any meaningful information but complicated implementation. It contains information about exceptions and deviations from general timetable. In the following part the pre-processing of the GTFS is described in more detail.

5.3.1 Processing of static GTFS data

As already mentioned, the GTFS data contains the complete timetable of public transportation in Switzerland. This includes buses, regional trains and even aerial cableway from many different public transportation agencies. Therefore, it was necessary to filter all files in the static GTFS and keep only those that are related to the SBB. For this the very handy python tool kit GTFSTK[9] has been used. GTFSTK is a tool kit developed to analyze and process GTFS data in memory without a database. It is based on python-based libraries pandas [10]and shapely[11]. For filtering the GTFS data the module "Feed" from the GTFSTK library has been used. Loading the static GTFS dataset into the Feed class creates an instance that represents a GTFS feed, for which multiple functions exist within the GTFSTK library. For filtering the method *restrict_to_routes()* has been used as summarized in figure 11.

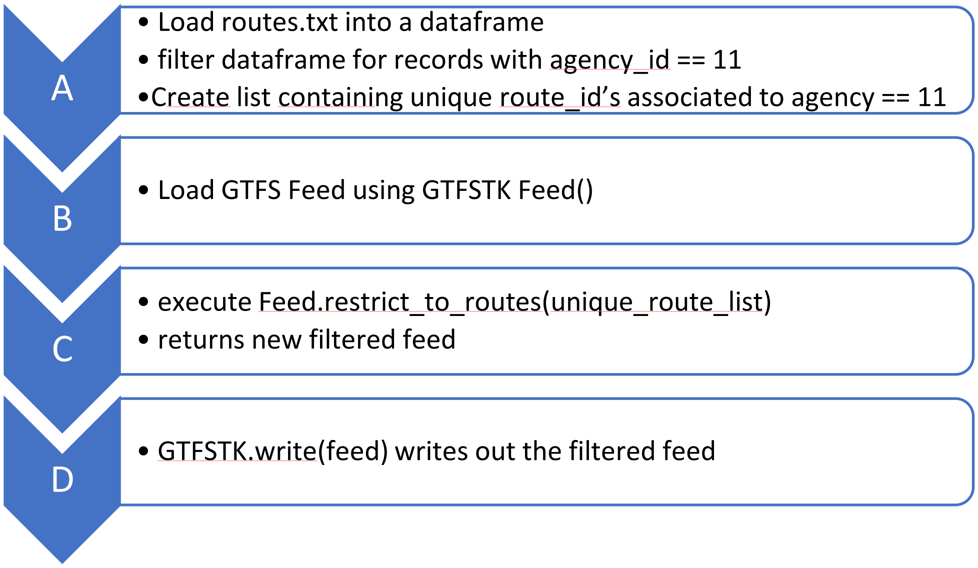

A
• Load routes.txt into a dataframe
• filter dataframe for records with agency_id == 11
•Create list containing unique route_id's associated to agency == 11
B
• Load GTFS Feed using GTFSTK Feed()
C
• execute Feed.restrict_to_routes(unique_route_list)
• returns new filtered feed
D
• GTFSTK.write(feed) writes out the filtered feed

Figure 11: Workflow for filtering a GTFS feed for all records related to one agency.

After the GTFS data has been filtered it was loaded into a graph-database[12]. For the purpose of this thesis it was important that the GTFS data could be queried in order to extract timetable information, such as a specific trip between two stations at a specific time. Van Bruggen (2015) proposed a graph model that allows to query GTFS data efficiently in a graph-database. This model has been adapted slightly to add the calendar-file, which is needed to query trips according to specific weekdays. The following figure illustrates the graph-model loaded into the graph database. Adaptions have been highlighted in orange. First, it was intended to compute the centrality measures (input features 21-33) within the graph-database. However, during the process of this thesis the idea evolved to compute the centrality measures time dependent as defined in table 2 and for this purpose the graph-database was not suitable because of limitations in process parallelization. Nevertheless, that graph-database was used for querying the GTFS data in order to calculate the centrality measures as described in the next section. Consequently, the calculation of the centrality measures could also be performed using a more common relational-database system.

[9] https://mrcagney.github.io/gtfstk_docs/index.html
[10] https://pandas.pydata.org/
[11] https://pypi.org/project/Shapely/
[12] https://neo4j.com

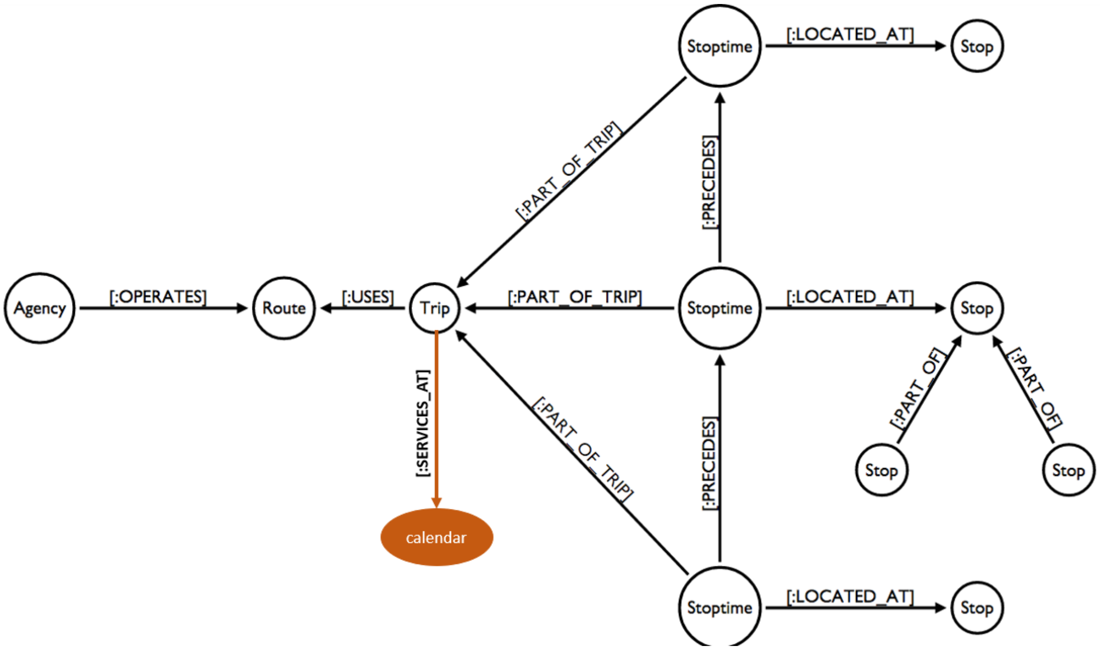


Figure 12: Schema used to model GTFS data in a graph-database. Proposed by Van Bruggen (2015) and modified for the purpose of this thesis (highlighted in orange).

5.3.2 Feature engineering GTFS

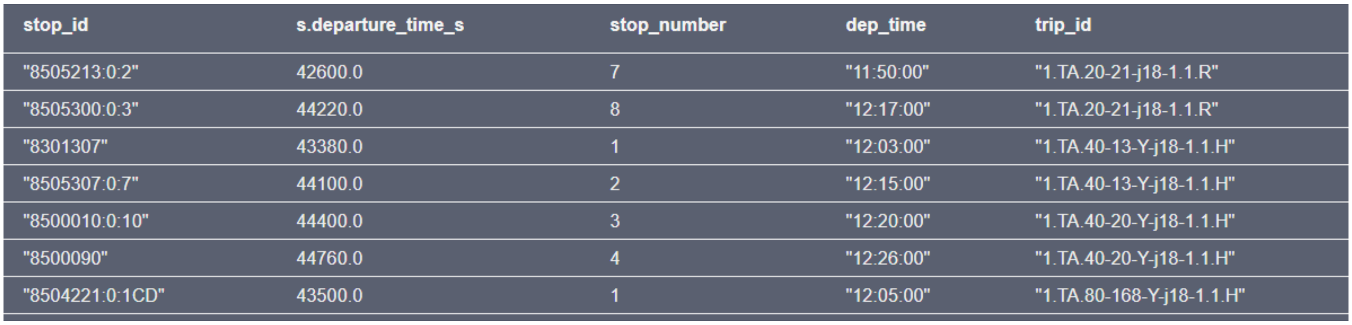
The calculation of the centrality measures are relied on the dataframe generated in section 5.2. This dataframe contains already each observation, for which the arrival delay should be predicted by the machine-learning model. This section describes how the centrality measures are calculated to receive the input features 18 and 20-33, which would finalize the input table to predict arrival delays.

Centrality measures are calculated by building a multi-directed graph based on timetable information. Within the graph stations are represented as nodes and train-services as connecting edges. As outlined in section 4.2, the centrality measures should capture the topological properties pre-defined by the provided timetable. Based on this graph the input features 18 and 20-33 are calculated. As defined in table 2 all centrality measures are calculated for the specific timeframe $[\boldsymbol{dt_i^v - 1.5h < t < dt_i^v + 1.5h}]$. For the definition of the centrality measures, it's worth considering the multi-directed graph $\boldsymbol{G}$ with $\boldsymbol{S}$ nodes and $\boldsymbol{C}$ connections. $\boldsymbol{G}$ is represents the graph within time $[\boldsymbol{dt_i^v - 1.5h < t < dt_i^v + 1.5h}]$ extracted from the timetable-database presented in the previous section. The extraction is performed using the following Cypher-query[13]:

```
Match(s: Stoptime)-[: PART_OF_TRIP]->(t:Trip)-[: SERVICES_AT(service:Service{day: "1"})
Where timedown < s.departure_time_s AND s.departure_time_s < timeup
Return s.stop_id as stop_id, s.stop_sequence as stop_number, s.departure_time as dep_time, t.id as trip_id ORDER BY trip_id,
stop_number ASC
```

[13] https://neo4j.com/developer/cypher-query-language/

The cypher-query extracts all trip, that have stoptimes within $[\boldsymbol{dt_i^v - 1.5h < t < dt_i^v + 1.5h}]$ (see "Where-Clause"). The parameter (highlighted in yellow) "timedown" corresponds to $\boldsymbol{dt_i^v - 1.5h}$, whereas "timeup" corresponds to $\boldsymbol{dt_i^v + 1.5h}$. The parameter "day" is equal to the weekday needed. This cypher-query allows to extract all scheduled trips for a specific timeframe and a specified weekday. The following sample shows the extraction of such a query for $\boldsymbol{t = 13{:}00}$, which corresponds to $\boldsymbol{43200\, seconds}$. The special cases around midnight have been accounted by using multiple if-else clauses to ensure, that trains leaving after midnight are also respected if $\mathbf{t < 24{:}00.}$ As emphasized GTFS does not specify the exact date on which a trip is executed. GTFS is based on a weekly pattern. This property of GTFS data has been used to calculate the centrality measures in a more efficient way. Instead of calculating the centralities for each observation in the input table, unique weekday $\boldsymbol{w}$ and train-service departure times $\boldsymbol{dt_x^v}$ pairs have been isolated from the dataframe previously generated. Each tuple of $(\boldsymbol{w, dt_x^v})$ was used to query the graph-database, thus each query-output has been used to build the corresponding graph $\boldsymbol{G}$ and caculate the centrality measures. As the calcuation of centralities is always computed for all nodes $\boldsymbol{S}$ involved in $\boldsymbol{G}$ one centrality-computation matches several observations in the input dataframe. Namely all observations with $(\boldsymbol{w, dt_x^v})$ and $\boldsymbol{s_i^v\ or\ s_j^v} \in \boldsymbol{S}$.



stop_id	s.departure_time_s	stop_number	dep_time	trip_id
"8505213:0:2"	42600.0	7	"11:50:00"	"1.TA.20-21-j18-1.1.R"
"8505300:0:3"	44220.0	8	"12:17:00"	"1.TA.20-21-j18-1.1.R"
"8301307"	43380.0	1	"12:03:00"	"1.TA.40-13-Y-j18-1.1.H"
"8505307:0:7"	44100.0	2	"12:15:00"	"1.TA.40-13-Y-j18-1.1.H"
"8500010:0:10"	44400.0	3	"12:20:00"	"1.TA.40-20-Y-j18-1.1.H"
"8500090"	44760.0	4	"12:26:00"	"1.TA.40-20-Y-j18-1.1.H"
"8504221:0:1CD"	43500.0	1	"12:05:00"	"1.TA.80-168-Y-j18-1.1.H"

Figure 13: Sample of extracted timetable information for t = 13:00. The first rows correspond to a trip that started earlier than 11:30, therefore the two first 5 legs of the trip are not included. The stop_id contains also platform information separated by ":", which has not been included for building the of the multi-directed graph. The column "s. departure_time_s" corresponds to the departure time of the trip in seconds.

The python package Py2neo[14] allows to access the graph-database directly within the python-script. Consequently, the extracted table as shown in figure 13 is used to build multi-directed graph $\boldsymbol{G}$ in python using the python graph-package NetworkX[15], which provides useful tools for the creation, manipulation and the study of the complex networks.

[14] https://py2neo.org/v4/

[15] https://networkx.github.io/documentation/stable/index.html

For better understanding of the calculation of the centrality measures a fictive simple multi-directed graph $\boldsymbol{GF}$ is shown in figure 14. The centrality measures have all been calculated using the algorithm module[16] provided by NetworkX, which contains built-in functions to calculate centrality measures of a graph. The centrality measures will be exemplified on the graph illustrated in figure 14, where arrows represent different train-services and nodes represent stations that are served by the corresponding train-service. As we calculate centrality measures for different timeframes, the centrality-measures should always be normalized according to its network-size. Otherwise, the centrality-measures between different networks, timeframes respectively, cannot be compared.

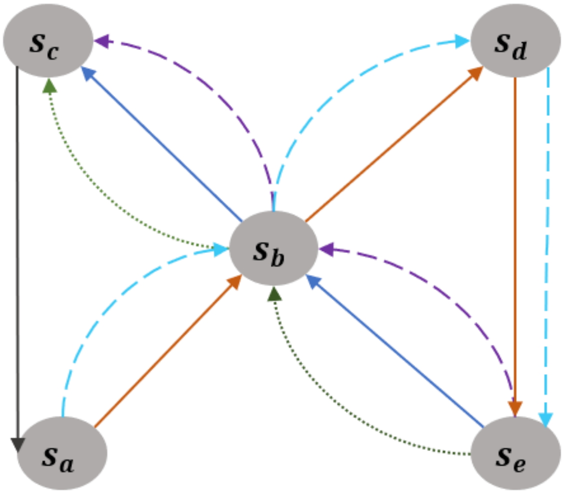


Figure 14: Example of a multi-directed-graph built after extracting the corresponding trips from the graph-database. The different colours indicate individual trips. Nodes represent stations

- Indegree- and outdegree- centrality

Normalized indegree centrality for node $\boldsymbol{s}$ is defined as (Marsden, 2015):

$$\boldsymbol{C_i(s) \ = \frac{1}{n-1} \sum_{j=1}^{n} x_{ij}}$$

Normalized outdegree centrality for node $\boldsymbol{s}$ is defined as (Marsden, 2015):

$$\boldsymbol{C_o(s) \ = \frac{1}{n-1} \sum_{j=1}^{n} x_{ji}}$$

Where $\boldsymbol{x}$ corresponds to the incoming and outgoing edges respectively. $\boldsymbol{n}$ corresponds the number of nodes in the network. Therefore, in graph (fig 12), indegree- and outdegree-centrality for node $\boldsymbol{s_a}$ would be:

$$\boldsymbol{C_i(s_a) \ = \ 0.25}$$

$$\boldsymbol{C_o(s_a) \ = \ 0.5}$$

[16] https://networkx.github.io/documentation/networkx-1.10/reference/algorithms.html

- Betweenness-, edge-betweenness- and load-centrality

For a set of nodes $\boldsymbol{S}$ betweenness-centrality is based on the number of shortest paths between two nodes $\boldsymbol{s_i, s_j} \in \boldsymbol{S}$ and denoted as $\boldsymbol{\sigma(s_i, s_j)}$. $\boldsymbol{\sigma(s_i, s_j \backslash v)}$ corresponds to the number of shortest paths going through node $\boldsymbol{v}$. The implementation of betweenness-centrality corresponds to the algorithm proposed by Brandes (2008), who defined betweenness-centrality as:

$$\boldsymbol{C_B(v)} \; = \sum_{s_i, s_j \in S} \frac{\boldsymbol{\sigma(s_i, s_j \backslash v)}}{\boldsymbol{\sigma(s_i, s_j)}}$$

Betweenness-centrality for node $\boldsymbol{v}$ normalized by the number of all shortest-paths between $\boldsymbol{(s_i, s_j)}$. Considering figure 14, we can see that a multi-directed graph can have multiple shortest path between for $\boldsymbol{(s_i, s_j)}$. The provided algorithm by Brandes (2008) has encountered this problem as following. In this case the number of shortest paths connecting $\boldsymbol{(s_i, s_j)}$ depends on the multiplicity of their edges: tripling an edge of a path is resulting in three different paths of the same length, as all copies of the tripled edge can be used. If more than one edge has multiplicity larger than one, then any instance of one edge combined with any instance of another edge yields a different path. Consequently, the total number of paths obtained from a generic path is the product of the multiplicities of its edges (Brandes, 2008). Same degree-centrality, betweenness-centrality is normalized using $\boldsymbol{1/((n-1)(n-2))}$ as suggested by Freeman (1979). For the graph in figure 14, $\boldsymbol{C_B(v)}$ for station $\boldsymbol{s_a}$ would be:

$$\boldsymbol{C_B(s_a) = 0.25}$$

As we can see in figure 14, $\boldsymbol{s_a}$ is not a very "central" station within the graph. For comparison betweenness-centrality for station $\boldsymbol{C_B(s_b) = 0.83}$, which seems plausible if look at athe number train-services, which pass $\boldsymbol{s_b}$ to reach another station. Edge-betweenness-centrality is very closely related to betweenness-centrality, it is a natural extension of betweenness to edges and can be obtained by replacing $\boldsymbol{\sigma(s_i, s_j \backslash v)}$ with $\boldsymbol{\sigma(s_i, s_j \backslash e)}$, that corresponds to the number of shortest $\boldsymbol{(s_i, s_j)}$-paths containing edge $\boldsymbol{e}$ (Brandes, 2008)**.** Load-centrality has often been misunderstood as equal to betweeneess-centrality (Brandes, 2008). The small difference between load and betweenness centrality is the way the calculation algorithm is implemented, which leads to slightly different results (Brandes, 2008).

- Closeness-centrality

Closeness centrality of a node $\boldsymbol{s_i}$ is the reciprocal of the sum of the shortest paths distances from $\boldsymbol{v}$ to all other nodes (Freeman, 1979). Closeness-centrality is defined as:

$$\boldsymbol{C_C(s_i)} = \frac{\boldsymbol{n-1}}{\sum_{i=i}^{n-1} \boldsymbol{d(s_i, s_j)}}$$

, where $\boldsymbol{d(s_i, s_j)}$ is the number of edges in the shortest-path connecting $\boldsymbol{s_i}$ and $\boldsymbol{s_j}$**.** For closeness-centrality there is no need for additional normalization, since the sum of shortests-paths depends on the number of nodes in the network. Closeness-centrality for $\boldsymbol{s_a}$ in figure 14, would be:

$$\boldsymbol{C_C(s_a) = 0.4}$$

As we can see from the result, the closeness-centrality algorithm does not make a distinction between, parallel edges in a multi-directed graphs such as betweenness-centrality. Nevertheless, the centrality measure can still be an indicator for its reachability. Therefore, a station with a high closeness degree, can be well reached from any other station in the network.

- PageRank

Pagerank ranks nodes in the the graph based on the structure of the incoming links. It is based on a recursive algorithm and was originally developed to rank documents and developed by Goolge-founders Brin and Page (1998). It represents the likelihood that a node can be reached by a random traveller within the network (To, 2015; Zhong *et al.*, 2015). To compute PageRank it was necessary to create a weighted graph $\boldsymbol{H}$ from graph $\boldsymbol{G}$. Using NetworkX this could be done very easily by counting the number edges $\boldsymbol{e}$ between two nodes and replacing them with a single edge with weight $\boldsymbol{sum}(\boldsymbol{e})$.

- Busy-Index

The busy-index not a classic centrality-measure. As already mentioned it is a simple count-function and corresponds to the weight $\boldsymbol{sum}(\boldsymbol{e})$ of two connected nodes $\in \boldsymbol{H}$.

In this section the preparation of the static GTFS data has been described and how it has been modelled within the graph-database to allow querying timtable information. Further, it is described how the station-related timetable features 18-33 (see table 2) have been calculated using the graph-database and the processed actual dataset. Consequently, the dataset is now finally ready processing within the the machine-learning algorithm. The final input tables contains 2’148’896 observations and should therefore correspond to the number of train arrivals within the long-distance traffic in Switzerland within the time-period 1.1.2018 – 30.6.2018.

The next section is dedicated to the machine-learning model and how it has been trained and validated to produce the results needed to investigate on the research questions. In addition, the datapipeline for feature-enconding and normalization will be discussed.

5.4 Machine-learning part

5.4.1 General machine-learning workflow

The prediction of arrival delay at the next three following stations can be accomplished by using three different models, which will called model_1, model_2 and model_3 in the following. These models perform the prediction tasks defined below:

$$\begin{aligned} &\text{model\_1:} \quad && \boldsymbol{\vartheta_1(t)} \rightarrow \boldsymbol{\hat{y}_j^v} \\ &\text{model\_2:} \quad && \boldsymbol{\vartheta_2(t)} \rightarrow \boldsymbol{\hat{y}_{j+1}^v} \\ &\text{model\_3:} \quad && \boldsymbol{\vartheta_3(t)} \rightarrow \boldsymbol{\hat{y}_{j+2}^v} \end{aligned}$$

In order to assess the effects of specific input feature categories as defined in research question 1 and 2 the procedure visualized in table 6 has been performed. Therefore, the three prediction models were first trained on all input features and validated using 10-Fold cross-validation. Afterwards, the first input feature category has been removed and the models were retrained using the remaining input features. Then the removed category was replaced by the next category. The only input feature category, that has not been removed were trip-related input features, because leaving them away would remove the context of railway transportation. In table (6) one can see the combination of input feature categories that have been tested. Consequently, five different input feature combinations have been tested and validated to predict $\boldsymbol{\hat{y}_j^v}$, $\boldsymbol{\hat{y}_{j+1}^v}$, $\boldsymbol{\hat{y}_{j+2}^v}$. These different input feature spaces are named after the feature category that has been left out for model validation.

Feature combination matrix	**all features**	**no delay propagation**	**no snapshot**	**no station-related**	**no network**
Trip-related	**X**	**X**	**X**	**X**	**X**
Delay propagation	**X**		**X**	**X**	**X**
Snapshot	**X**	**X**		**X**	**X**
Station-related	**X**	**X**	**X**		**X**
Network-related	**X**	**X**	**X**	**X**	

Table 6: Shows the different combination of input feature categories contained by each input feature space. In sum five different input feature spaces will be evaluated using 10-Fold cross-validation for each prediction model ϑ_1, ϑ_2, ϑ_3.

The five different input feature spaces and the three different prediction models results in 15 cross-validation performances. In order to meet this computational challenge a virtual-machine (VM) on Microsoft's Azure[17] platform has been used in addition to a Lenovo Thinkpad E580, InterCore i7 with 32 GB RAM. The VM used on Azure had 8 Cores with 56 GB RAM, which was also used to deploy the graph-database described in section 5.3.2. For cross-validation the python package scikit-learn[18] has been used, which also allows to calculate the following measures for model evalutation:

[17] https://azure.microsoft.com

[18] https://scikit-learn.org/

- Mean absolute Error (MAE) $=\frac{1}{n}\sum_{i=1}^{n}|\widehat{\boldsymbol{Y}_i}-\boldsymbol{Y}_i|$

- Mean squared Error (MSE) $=\frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{Y}_i-\widehat{\boldsymbol{Y}_i})^{\boldsymbol{2}}$

- Root Mean Squared Error (RMSE) $=\sqrt{\frac{\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}(\boldsymbol{Y}_i-\widehat{\boldsymbol{Y}_i})^{\boldsymbol{2}}}{\boldsymbol{n}}}$

- Coefficient of determination R^2 $=\boldsymbol{1}-\frac{\sum_{i=1}^{n}(\boldsymbol{Y}_i-\widehat{\boldsymbol{Y}_i})^{\boldsymbol{2}}}{\sum_{i=1}^{n}(\boldsymbol{Y}_i-\overline{\boldsymbol{Y}_i})^{\boldsymbol{2}}}$

These error measures are commonly used to evaluate the prediciton accuracy of a model (Pongnumkul *et al.*, 2014; Kecman and Goverde, 2015; Čelan and Lep, 2017; Lessan, Fu and Wen, 2019). MAE measures the absolute residual between the prediction value $\widehat{\boldsymbol{Y}_i}$ and the true value $\boldsymbol{Y}_i$ for each observation and calculates the sum of it, which then is divided by the number of observations. Thus, the resulting MAE is the undirected typical magnitude of the residual. Consequently, MAE is a precision measure and scale-dependent and should be as small as possible (Hyndman and Athanasopoulos, 2018). MSE calculates the sum of the squares of the residuals, thus it incorporates the variance of the prediciton model. RMSE is the root of MSE and therefore has the same unit as the predicted values and measures the error rate of a model. R^2 summarizes the explanatory power of a regression model. It describes the proportion of variance in the dependent variable that can be explained by the regression model (James *et al.*, 2013; Hyndman and Athanasopoulos, 2018).

As mentioned in the approach-describtion of research question 1 the use of a tree-based machine-learning algorithm allows to extract the feature importance of each input feature. The feature importance analyse is focusing on the model that uses all features, because this way we can identify dominant features among the whole input feature space. In the other models feature importance my be biased by the absence of other features. A possible approach to analyze feature importance would be to train and test a model using a training and test dataset and afterwards extract the feature imortances of the model. However, this approach may not address the variety in the data and was therefore considered as unsatisfying. Instead, a more systematic approach has been chosen that has been performed using K-Fold cross-validation. The method *cross_validate()*[19] from the python package scikit-learn allows to return the fitted model after each fold during K-Fold cross-validation. Usig this method it is possible to extract the feature importance of each feature for each fold within K-Fold cross-validation. Thus, for each feature 10 feature importance measures could be extracted for the models $\boldsymbol{\vartheta_1}, \boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$.

[19] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html

5.4.2 Selecting the machine-learning algorithm and hyperparameters

In order to adress the first research question a specifc machine-learning algorithm has been chosen. XGBoost[20] which is an implementation of Gradient Boosting Decision Tree (GBDT) algorithm has been widely recognized in a number of machine-learning and data mining challenges (Yamaguchi, As and Mine, 2019). A benefit of using ensembles of decision tree methods like gradient boosting or RandomForest is that provide estimates of feature importance from the trained predictive model. The basic idea of GBDT is combining a series of weak base classifieres into a strong one (Rao *et al.*, 2019). Gradient boosting builds trees one after one, where each new tree helps to correct errors made by the previously trained tree, consequently it is a forward, stagewise procedure. For regression problems boosting is a numerical optimization technique. It refers to the process where a loss function is minimized by reducing the residuals of the aforegoing tree. Consequently, for regression the first regression tree is the one that, for the selected tree size, maximally reduces the loss function. Each following regression tree aims to reduce the residuals (Elith, Leathwick and Hastie, 2008). In XGBoost the default loss-function for regression tasks is squared-error loss function (XGBoost, 2019). In the previous section the extraction of feature importance has been described. Tree-based split their population using lables into homogenous sets based on the most important variable. This is decided by Gini index to evaluate the quality of a a particular split. The Gini index is defined on the impurity of a node, which denotes, how homogenous the values within the node are (James *et al.*, 2013). Based on this, XGBoost calculates the feature importance measure “gain”. Gain is defined as the improvement in accuracy brought by a feature to the branches it is on (XGBoost, 2019). Therefore, the gain-score of a feature is the average gain across all splits the feature is used in. This corresponds to the relative importance of a feature within GBDT (Hastie, Tibshirani and Friedman, 2009). The secon measure provided by XGBoost is “cover”. Cover measures the relative quantity of observations concerned by a feature, which means it measures how many observations are affected if the feature is used for splitting a tree (XGBoost, 2019).

Among other GBDT implementation XGBoost has been chosen is its scalability in all scenarios. The scalability of XGBoost is due to several important systems and algorithmic optimizations. Further XGBoost exploits out-of-core computation and enables data scientists to exlpoit a large amount of easily on a dekstop (Chen and Guestrin, no date). Furthermore, regarding delay prediction in bus transportation services, it has been shown that XGBoost slightly outperforms other common machine-learning algorithms such as ANN, SVR, linear regression and RandomForest (Yamaguchi, As and Mine, 2019).

XGBoost comes along with different hyperparameters, which need to be tuned. However, hyperparameter-tuning is a computational very expensive operation as for each hyperparamter to e tuned a K-Fold cross-validation is performed using GridSearchCV[21] or RandomizedSearchCV[22]. For this thesis hyperparameter-tuning was performed as proposed by Jain (2015). The final hyperparameters are listed in table 7 .

[20] https://xgboost.readthedocs.io/en/latest/

[21] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

[22] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

Parameter	Values
objective	reg:squarederror
learning_rate	0.058
max_depth	6
gamma	0.015
n_estimators	500
colsample_bytree	0.921
subsample	0.830

Table 7: Hyperparameter used in XGBoost. The first parameter denotes to perform a regression using squared-error loss function.

5.4.3 Datapipeline

A datapipeline ensures that the input data is always prepared equally, before processing in the machine-learning algorithm. Nominal features (1-8 in table 2) have been factorized using the python library Pandas[23] and its factorize-method[24]. The method is useful to obtain a numeric representation of nominal data columns. As the centrality-measures needed to normalized in order to account network-size effects, all other features also have been normalized. This should ensure that the centrality-measures that range within 0 and 1 are negelected by the machine-learning algorithm. This has been by using the scikit-learns MinMax-Scaler[25], which translates each feature individually, such that it is representable within a range between 0 and 1 as the centrality-measures. Input features that represent time, have been normalized seperately. Due to the repetitiveness in railway-networks, input features representing dates and time should be considered as cyclical features (Kaleko, 2017). The solution is to split the date and time features using cosine and sinus values to maintain the cyclical property of the features (Kaleko, 2017). In addition the datapipeline treats NA-values, for example for the special cases mentioned in section (Input feature space)

5.4.4 Learning curve

In order to account research question 3 a learning curve will be produced. As learnining curve is produced by the systematic increase of training datasize. For each training datasize a 10-fold cross-validation will be produced, for which the training- and test-score will be returned. The scores are stored seperatly togehter with the corresponding training datasize. Consequently, after repeating this procedure and steadily increasing training datasize, one can plot the training- and test-score against the training datasize. Therefore, the plot visualizes how much the model profits by increasing the training datasize. In addition, a learining curve allows further interpretation of the model, regarding overfitting and model limitations.

[23] https://pandas.pydata.org/

[24] https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.factorize.html

[25] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

6. Results

The results chapter is structured as follows. First the results from computing the input feature space are presented, where a special focus lies on the results of the centrality measures, as these have been a central topic within this thesis, especially for the second research question. Afterwards each input feature class is put in relation to the arrival delays $\boldsymbol{y}_{j}^{v}$, $\boldsymbol{y}_{j+1}^{v}$ and $\boldsymbol{y}_{j+2}^{v}$ in order to investigate on their relation. For this different correlation plots will be presented. The second part of the results chapter will present the results from the delay prediction system and how the different input feature categories influence the prediction accuracy measured by the metrics presented in the previous chapter 5.4.1. This will be followed by the results derived from the feature importance analysis. In the third part the learning curve will be presented and discussed, which is the basis for research question 3.

6.1 Input feature space

6.1.1 Station-related features derived by centrality measures

The calculation of the input features was very intensive and time-consuming. In order to get a brief overview of the centrality measures a map has been created showing the location of the station. The size of the point is proportional to the centrality measure. The centrality measures for these maps have been calculated seperately in order to get an overview and a better understanding about these measures. These centrality measures have been calculated using all the data within graph-database, which means all trips have been included. In figure 15 one can see the visualization of the centrality measure “closeness”. Futher, it is visible that the stations within the city-triangle

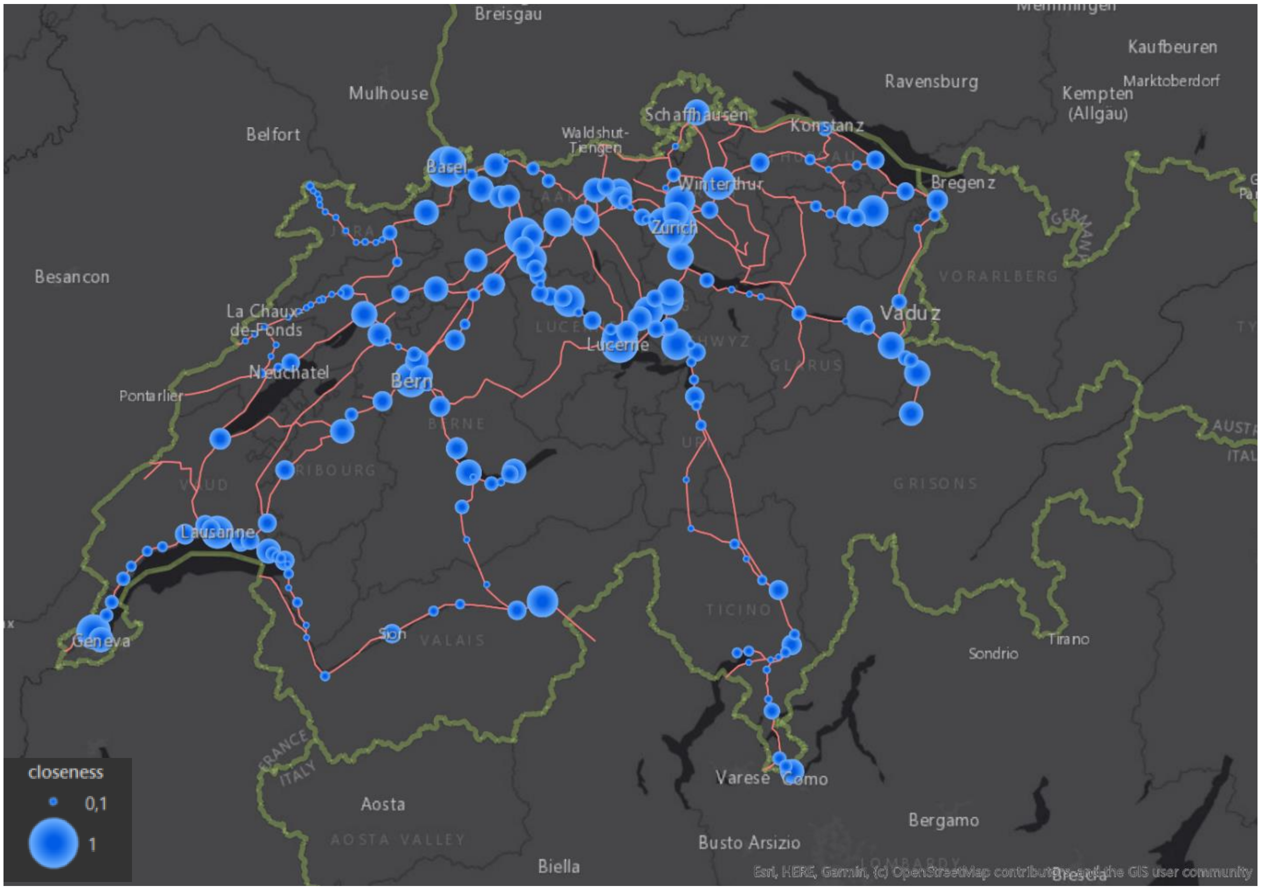


Figure 15: Closeness mapped for each station using the whole GTFS dataset.

“Zurich-Olten-Luzen” have all a relative high closeness measuers. An interesting fact is that in the Italian part of Switzerland, closeness-centrality is rather low for most stations except for the three major cities in these regions (Chiasso, Lugano and Bellinzona). This can be explained by the ICE and EC routes going from Basel to Mailand and and stop in Chiasso, Lugano and Bellinzona, whereas the other stations in this regions are not served by this route. Consequently, more edges must be passed to reach other nodes which reults in lower closeness-centrality. The same situation can be observed in the Canton of Wallis in the south-west. In this case Brig is an important station, connecting the region with other regions by InterCity lines 6 and 5 (see figure 1). Betweenness-centrality (showed in fig 16) measures the criticality of station by identifying all shortest-path connections between all nodes in the network. Unsurprisingly, Zurich has a very high betweenness-centrality as it is the main-station for Switzerlands railway traffic. Further, we can see the importance of a very special station, namely Arth-Goldau, which is located at the east of Lucerne. Arth-Goldau is an important node for the long-distance traffic system as it servers as a transfer-station for passenger going south or coming from south. Otherwise, the major cities as Bern, Lucern, Basel, Biel, Olten and Lausanne have high betweenness-centrality measures.

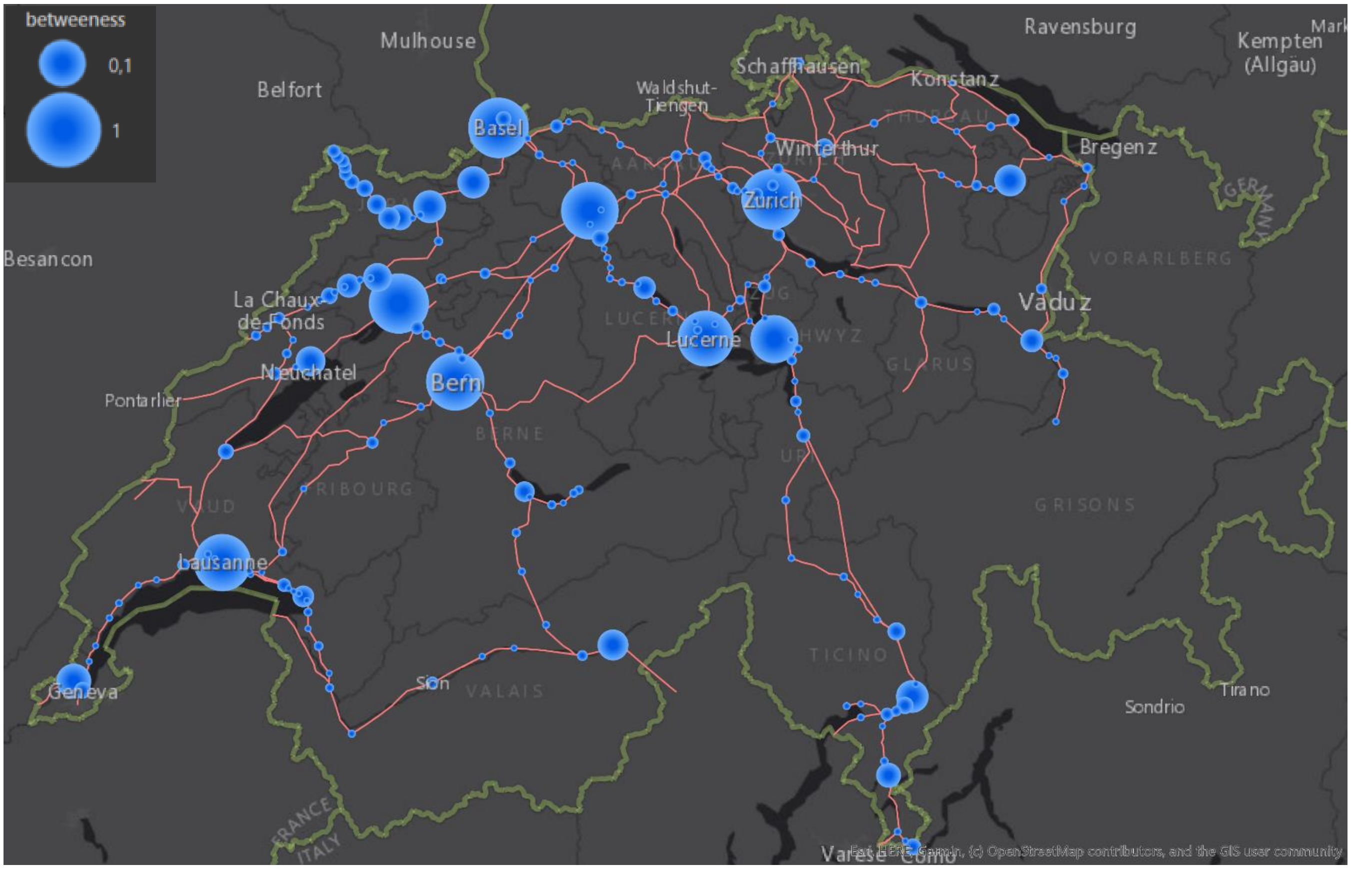


Figure 16: Betweenness-centrality mapped for each station using the whole GTFS dataset.

As we are interested in the relation between arrival-delays and centrality-measures the following plots (figure 18) show the relation of each centrality measure with arrival-delays using scatterplots. For this, the centrality measures of the that leg-destination station have been visualized together with the corresponding arrival delay of the observation. The data for these plots is derived from the final input dataset derived after the processing steps described in section 5.3.1 and 5.3.2. Therefore, these scatterplots contain around 2 million points. Betweenness-centrality varies between 0 and 0.357, whereas closeness-centrality varies between 0 and 0.203. Still, betweenness-centrality seems to have more values close to 0 than closeness centrality. Compared to normalized betweenness-centralities found for 28 different metro-systems by Derrible (2012) these values seem plausible. Compared to the centrality-measures found by Tu et al. (2013), the values found here are slightly higher but still within a plausible range. Nevertheless, both centrality-measures do not show any clear correlation between them and arrival delay. Edge-betweenes behaves to arrival delay very similar as betweenness-centrality. PageRank shows the smallest values for all arrival delays overall. Surprisingly, it seems that PageRank achieves only larger values if arrival delays are very small. Edge-betweenness and PageRank have in common that both measures have an outlier. For PageRank the oulier has a score 0.65 and edge-betweenness is 0.5. After some investigations, these outliers represent the measures for leg-destination Geneva after midgnight. Multiple train-service comming from Geneva-Airport serve Geneva and continue toward Lausanne. In figure 18 the scatterplots for the indegree-, outdegree- and load-centrality compared to arrival delays are compared. Same as for the previous discussed centrality measures, no clear relation between arrival-delays and the centrality measures can be identified. These findings have been confirmed by computing a correlation heatmap between the station-related centrality features and arrival delay. In figure 19, the left correlation plot shows the correlation among destination-station and the right one among origin-stations. Overall it can be stated that station-related input features that capture the topological timetable relations among stations show no linear relationship with arrival delays of train-services.

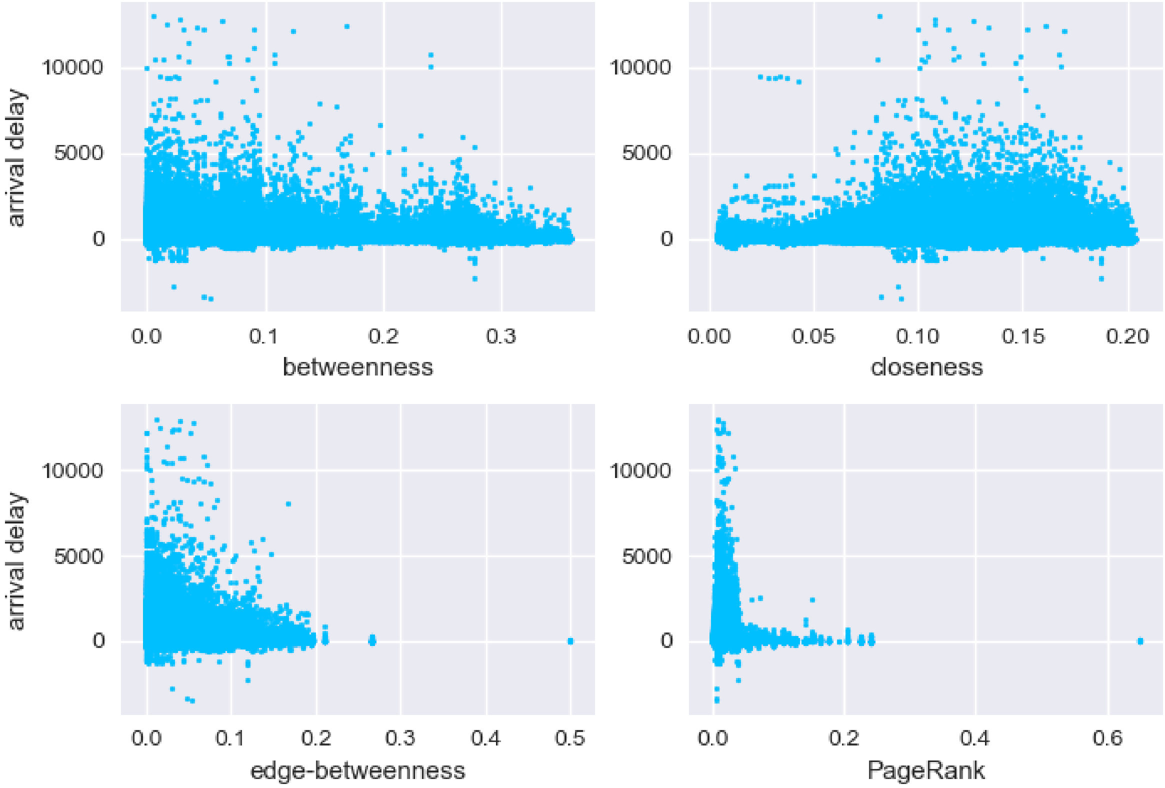
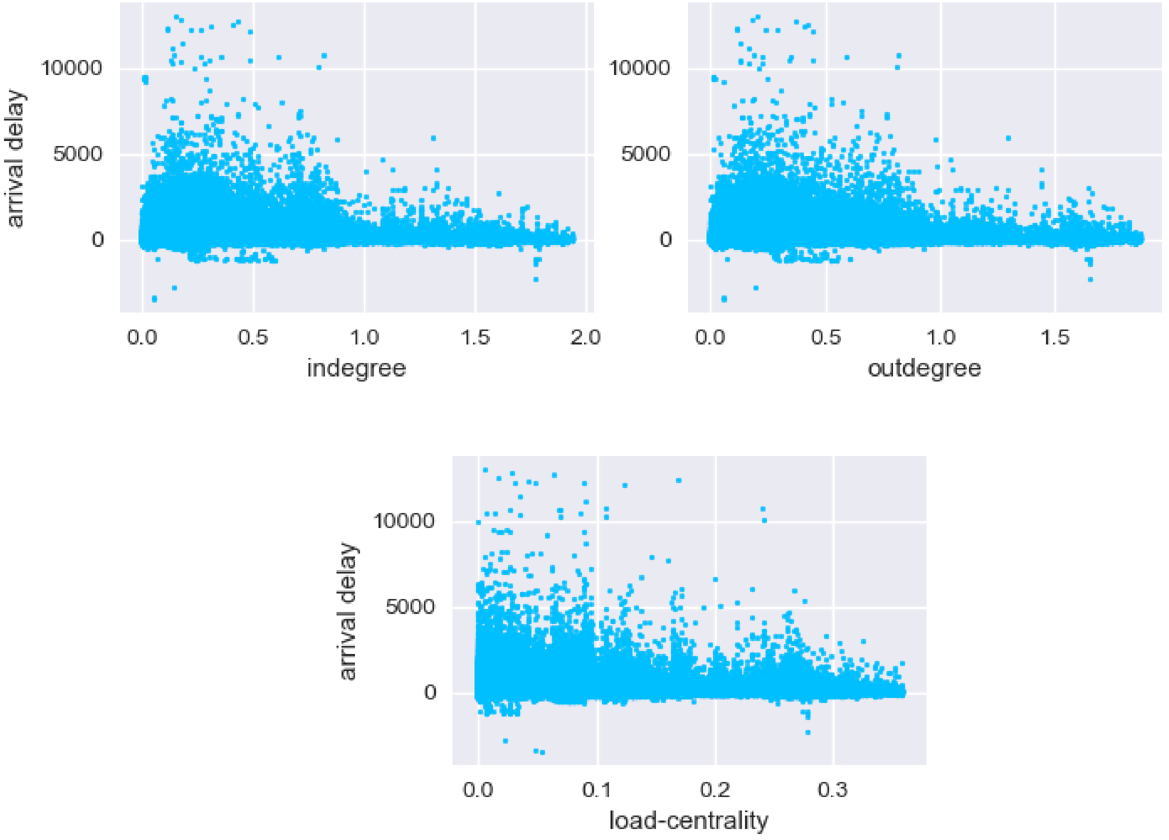


Figure 17: Scatterplots showing the relation between four centrality measures and arrival delay [sec]

Figure 19: Scatterplots showing relation between indegree-, outdegree- and load-centrality with arrival delays



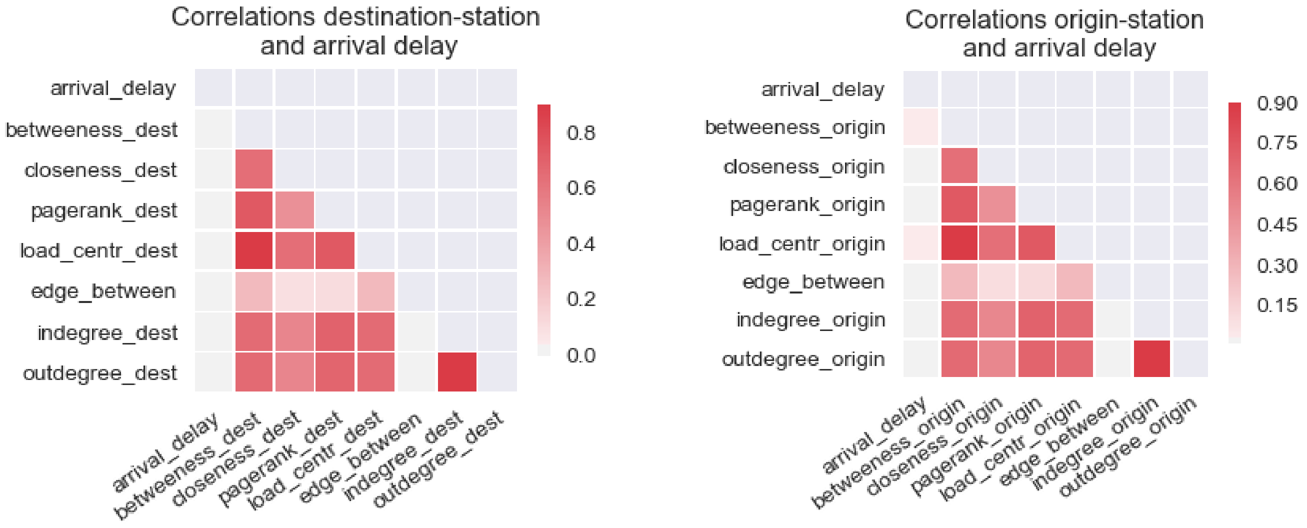


Figure 18: Correlation plots between station-related centrality features and arrival delay.

6.1.2 Snapshot, delay-propagation and network-related features

In this section the remaining input feature categories are analyzed regarding correlation with arrival delays and among each other. In figure 20, one can see the correlations between the features from the following input feauture categories: Network-related, snapshot and delay propagation features. Furthermore, the arrival delays of the stations $\boldsymbol{s_j^v}$, $\boldsymbol{s_{j+1}^v}$ and $\boldsymbol{s_{j+2}^v}$ are included. The strongest correlation with *arrival delay* is found with the input feature *current-delay*. This is not very surprising, considering that it seems plausible that a delayed train-service might also delayed at the following station. Therefore, it is also plausible that the input features *prev_delay* and *pre_prev_delay* also correlate with each other. Interesting to see is that *busy_index* does not correlate with *arrival_delay*. Consequently, a scheduled high frequency between two stations is not inducing delayed train services at the arrival station. *Snap_delay_prev_trip* and *snap_delay_last* correspond to the delay of the list trip of the same route $\boldsymbol{l^{v-1}}$ and the delay of the last train-services serving the same two subsequent stations. These two features also correlate with arrival delay but not as strong as the features from the delay-propagation category. Finally, *delays_at_station* is also a input feature that correlates with arrival delays. In figure 20 it is also visible that the correlation between *current_delay* and arrival delays of $\boldsymbol{y_{j+1}^v}$ and $\boldsymbol{y_{j+2}^v}$ declines. This is not unsurprising as time gap between the *current_delay* and $\boldsymbol{y_{j+1}^v}$ and $\boldsymbol{y_{j+2}^v}$ increases and delayed train-services have the possibilities catch up time or the other way around an unexpected disruption occurred. On the other hand, the snapshot feature category and network-related features (besides *busy-index*) are not affected by this decline.

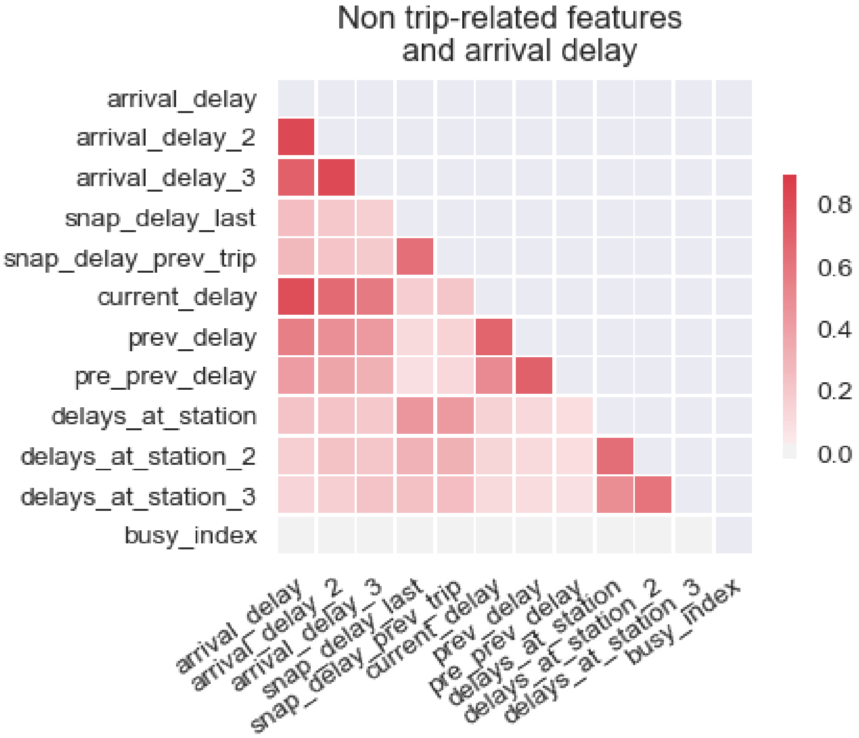


Figure 20: Features from categories: Network-related, snapshot and delay-propagation.

6.2 Results of the train delay prediction

In this section the results from the delay prediction model are presented. The results are presented within three tables. The results are derived after 10-Fold cross-validation on the whole dataset. Each table corresponds to the predictions of the models $\boldsymbol{\vartheta_1}, \boldsymbol{\vartheta_2}$and $\boldsymbol{\vartheta_3}$which predict the arrival delay for the three next stations as illustrated in figure 7. The prediction results have been assessed using the accuracy measures presented in section 5.4.1. Table 8, shows the results of prediction model $\boldsymbol{\vartheta_1}$. We can see that the best results are derived by using all input features available, as MAE, RMSE and MSE are the lowest. MAE measures the mean absolute error and has the same unit as the prediction variable and all individual prediction errors a weighted equally. Therefore, model $\boldsymbol{\vartheta_1}$ predicts the arrival delay $\boldsymbol{y_j^v}$ with a mean error below one minute. The same applies to the model by using different combination of input feature categories except for the combination "no delay propagation". This category includes the input features *current-delay*, *previous-delay*, and *pre-previous-delay*. As we have seen in the previous section, these are the features showing the highest correlation with the arrival delay of a train-service.

Model 1: predicts $\boldsymbol{y_j^v}$					
Feature group	all features	no station-related	no snapshot	no delay-propagation	no network
MAE	50.79	51.17	51.42	77.12	51.02
RMSE	107.64	108.23	108.04	171.10	107.58
MSE	11586.32	11712.87	11673.14	29276.51	11579.88
$\boldsymbol{R^2}$	0.67	0.67	0.67	0.18	0.67

Table 8: Presents the model evaluation measures for the prediction model 1. MAE represents the mean absolute error of the prediction in seconds. RMSE is the root mean square of the predictions also in seconds. MSE denotes the mean square error and R^2 is the coefficient of determination.

MSE is stable around 11'000 except for the model without delay propagation features. A high MSE value is not desirable, however as MSE squares the error of a prediciton value large prediction values fall heavily in weight. This might explain that the MSE value for the model without delay-propagation features is more than double compared to the others. RMSE is the square root of the MSE and has the same unit as MAE. RMSE is higher than MAE, which is unsuprising as the large prediction errors fall heavier into acount for its calculation. $\boldsymbol{R^2}$ is the proportion of the squared errors that can be explained given the input. $\boldsymbol{R^2}$ asseses the goodness of fit of the underlying regression function (Oruganti *et al.*, 2016). We can see that $\boldsymbol{R^2}$ also stable at 0.67 except for the model without delay-propagation features. All in all we can see that the input feature category combinations do not affect the results of the prediction, except if delay-propagation features are not included. Compared to other findings (Wang and Work, 2015; Oneto *et al.*, 2018) the results presented here are good. However, it cannot be stated that the model is better, as the data is different on the one hand. On the other hand, the different railway network settings may also contribute to different results. In figure 21 we can see the predicted arrival delays in relation to the true values of arrival delays. Herefore, the model has been trained using all input features on around 1.5 million observations to predict on the unseen test dateset. The result is similar to those derived after 10-Fold cross-validation. In the plot we can see that the model predicts high arrival delays, whereas the true value is null. After some investigation the following scenario could be detected that could cause this problem. According to the train-decriber some inconsistencies could identified. For example, a train that was heavily delayed at the previous station, was suddenly on time on the following station. This could be explained by re-scheduling operations by the railway

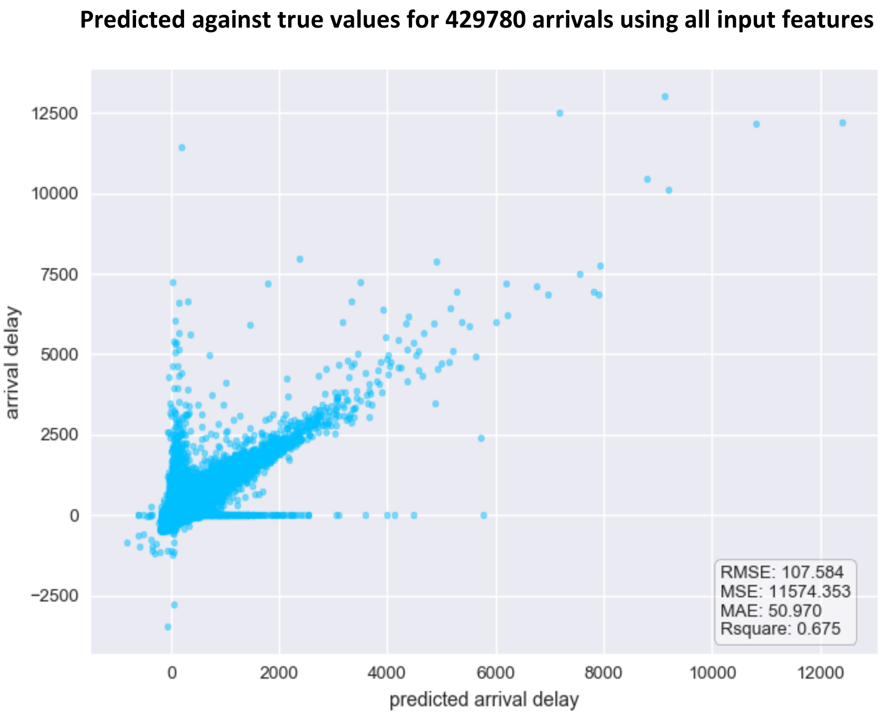


Figure 21: Scatterplot of predicted arrival delays and true values.

operator, like diversions. Consequently, the train-service gets a new schedule according to the current traffic situation and therefore is on time again for the new schedule and the registered arrival delay is 0. But the registered arrival delays of the previous stations, which correspond to *current-delay*, *prev-delay* and *pre-prev-delay* are still large. On the other hand the model predicts no delay, where in truth a large delay occurs. This shows clearly a limitation of the prediction model in emphasizing initial delays caused by unexpected disturbances, such as accidents weather conditions or other malfunctions.

Model 2: predicts y_{j+1}^{v}					
Feature group	all features	no station-related	no snapshot	no delay propagation	no network-related
MAE	54.24	54.54	55.37	78.0	61.73
RMSE	118.34	118.73	121.39	171.26	127.19
MSE	14004.74	14097.56	14735.95	29433.63	16176.84
R^2	0.55	0.55	0.54	0.18	0.55

Table 9: Results after 10-Fold cross-validation of model 2.

The prediction model $\boldsymbol{\vartheta}_2$, which predicts the arrival delay of the station after the next station and has achieved the results summarized in Table 9. We can see an increase of RMSE and MAE, which is explainable as the value to be predicted lies further in the future. Consequently, the proportion of the squared errors that can be explained given the input is declining. Contrary, to table 8 we can observe a difference between different input feature category combinations. While leaving out station-related and snapshot features does not result into an increase of prediction errors, leaving out network-related features has led to increasing prediction errors, registered with higher RMSE and MAE values. As we could see in figure 20, the correlation between input features *delays_at_station*, *delays_at_station_2* and *delays_at station_3* and the corresponding arrival delays do not decline, while *current_delay*, *prev_delay* and *pre_prev_delay* do decline from *arrival_delay* to *arrival_delay_3*. Consequently, the input features *delays_at_stations*, which measure the average arrival delay at stations $\boldsymbol{s_j^v}$, $\boldsymbol{s_{j+1}^v}$, $\boldsymbol{s_{j+2}^v}$ during the last three hours are useful input features for the predictions of arrival delays that lie further in the future. Considering table 10, we can identify the same behaviour for model 3. Similar to table 9, the prediction accuracy is declining, registered with increasing values for MAE, RMSE and MSE. Further, the model's ability to explain the variance within the data is declining. As in table 9, we can see that the inclusion of network-related features softens the increasing of the prediction error for events that are further in the future.

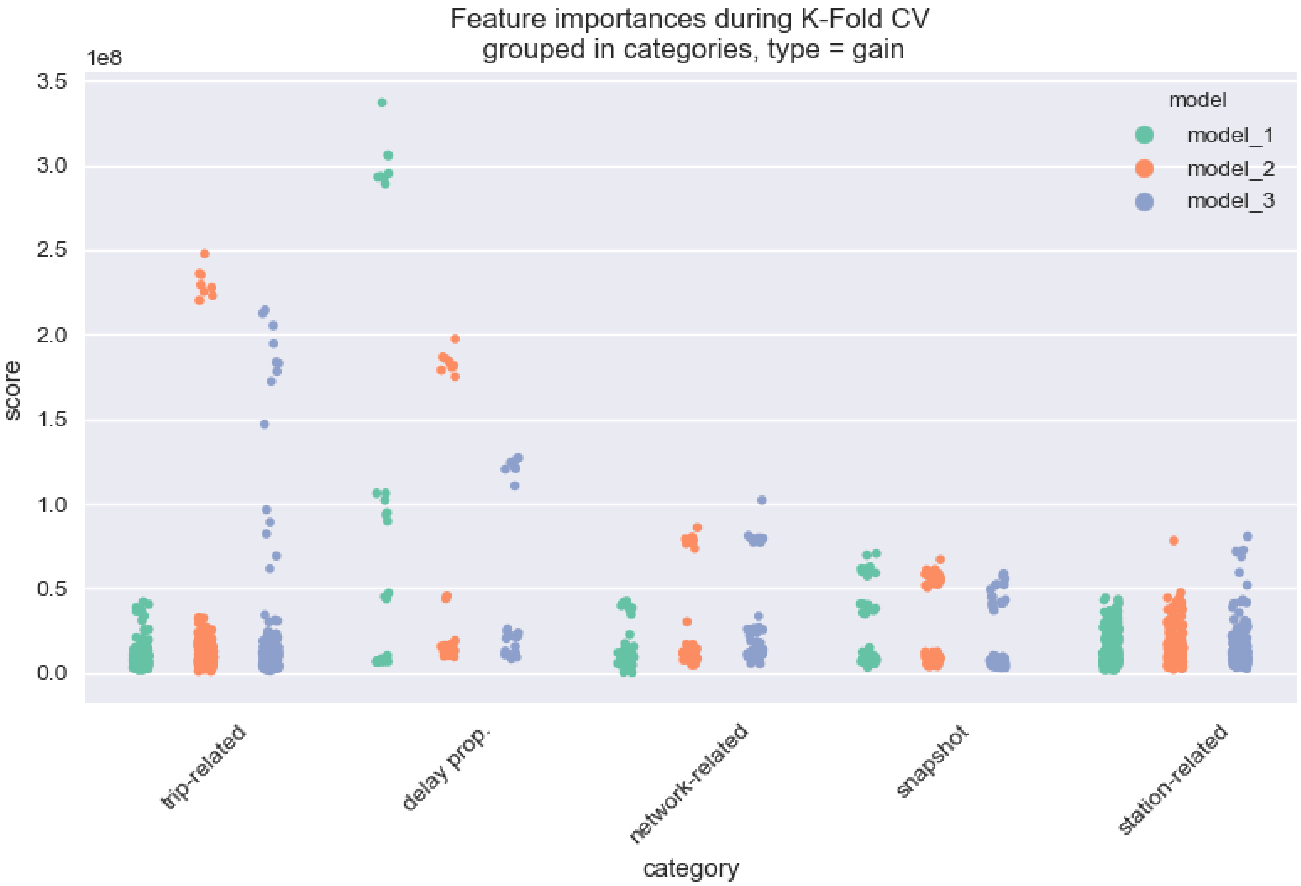
Model 3: predicts $\boldsymbol{y_{j+2}^v}$					
Feature group	all features	no centrality	no snapshot	no delay propagation	no network
MAE	52.07	52.22	52.20	78.26	67.36
RMSE	121.69	122.04	122.25	171.39	139.79
MSE	14809.15	14892.96	14945.26	29374.50	19540.48
$\boldsymbol{R^2}$	0.46	0.46	0.45	0.18	0.45

Table 10: Results after 10-Fold cross-validation of model 3.

6.2.1 Feature importance measure

During the 10-Fold cross-validation the feature importances for each fold have been extracted as described in section 5.4.1. The results are shown in figure 22. The feature importance has beeen measured using the importance measure 'gain' implemented in XGBoost. Gain is defined as the improvement in accuracy brought by a feature to the branches it is on (XGBoost, 2019). In figure 22 the different colors emphasize the model ($\boldsymbol{\vartheta_1}, \boldsymbol{\vartheta_2}$and $\boldsymbol{\vartheta_3}$). The groups on the x-axis correspond to the different input feature categories. We can see that using delay-propagation features for splitting in the decision trees results in the highest accuracy improvement and thus in the highest relative feature importance scores. The overall relative feature importance for delay-propagation features is decreasing for $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$. On the other side, a tendency of increasing feature importance for network-related features from model $\boldsymbol{\vartheta_1}$ to model $\boldsymbol{\vartheta_3}$ was registered. Consequently, we can see that network-related features, especially feature 18 (see table 2) are useful features to include, when predicting train arrival delays that are beyond the direct following station. However, they still do not reach the relative feature importance of delay propagation features. Snapshot features are more or less on stable niveau around 0.5*10^8. The snapshot feature category shows segregatted behaviour, while the input feature *snap_delay_last* has a higher relative feature importance than its compatriot *snap_delay_prev_trip*. This seems plausible as the time interval between the train-service $\boldsymbol{l^v}$ and $\boldsymbol{l^{v-1}}$ might be larger than the time interval between $\boldsymbol{l^v}$ and any other service $\boldsymbol{l^x}$. A further insight is that some trip-related features show higher importance scores for model $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$. In combination with figure 24, showing the top 25 averaged feature score during 10-Fold cross-validation for model $\boldsymbol{\vartheta_2}$, we can see that the trip-related feature *stop_number_2* achieved high feature scores. In model $\boldsymbol{\vartheta_3}$ the features *stop_number_2* and *stop_number_3* correspond to the leg-origin station $\boldsymbol{s_{j+1}^v}$ and leg-destination station $\boldsymbol{s_{j+2}^v}$ and were also trip-related features that achieved high gain-scores (fig. 25). Further, in figure 24 and figure 25 it is visible that closeness-centrality is the station-related feature with the highest gain scores, followed by betweenness and PageRank respectively. Despite having no visible relationship with arrival delays they seem to contribute within the prediction model.

Figure 22: Strip-plot of relative feature importance during 10-Fold cross-validation.



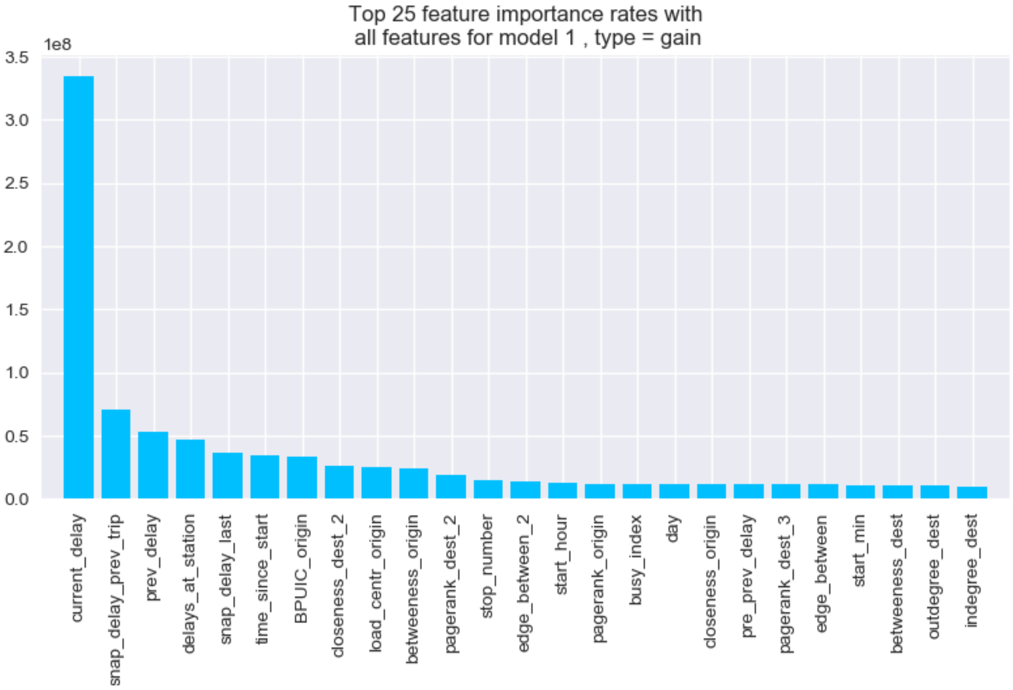


Figure 23: Top 25 feature importance scores during K-fold CV. Each feature score has been averaged over all folds

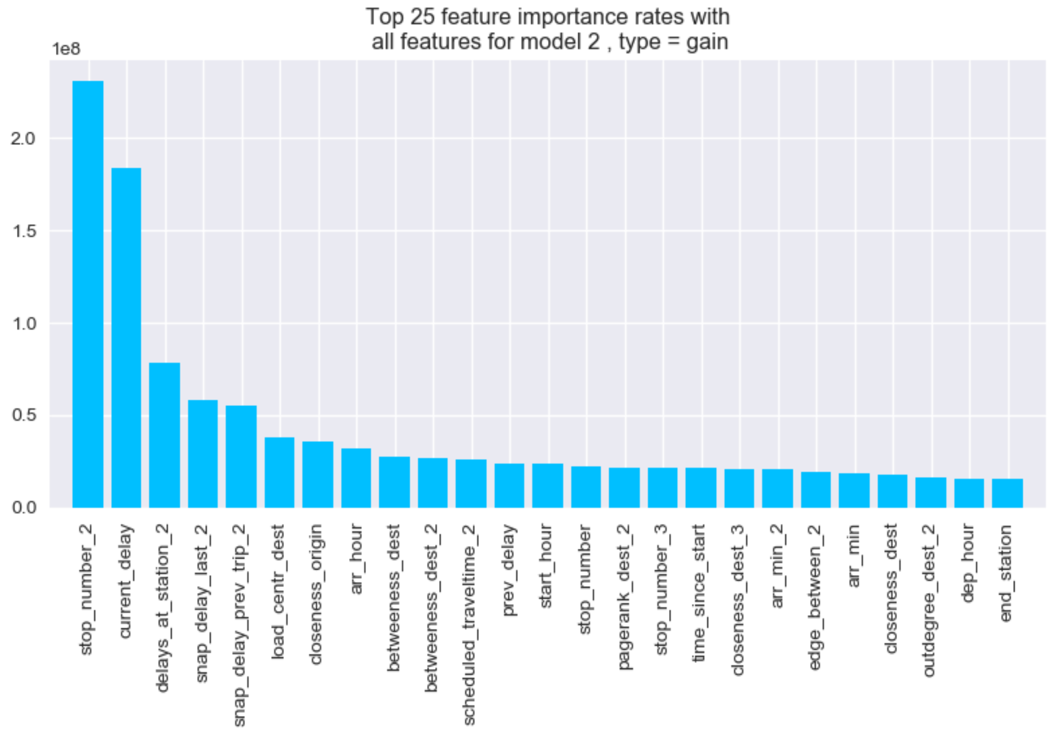


Figure 24: Top 25 feature importance scores during K-fold CV. Each feature score has been averaged over all folds

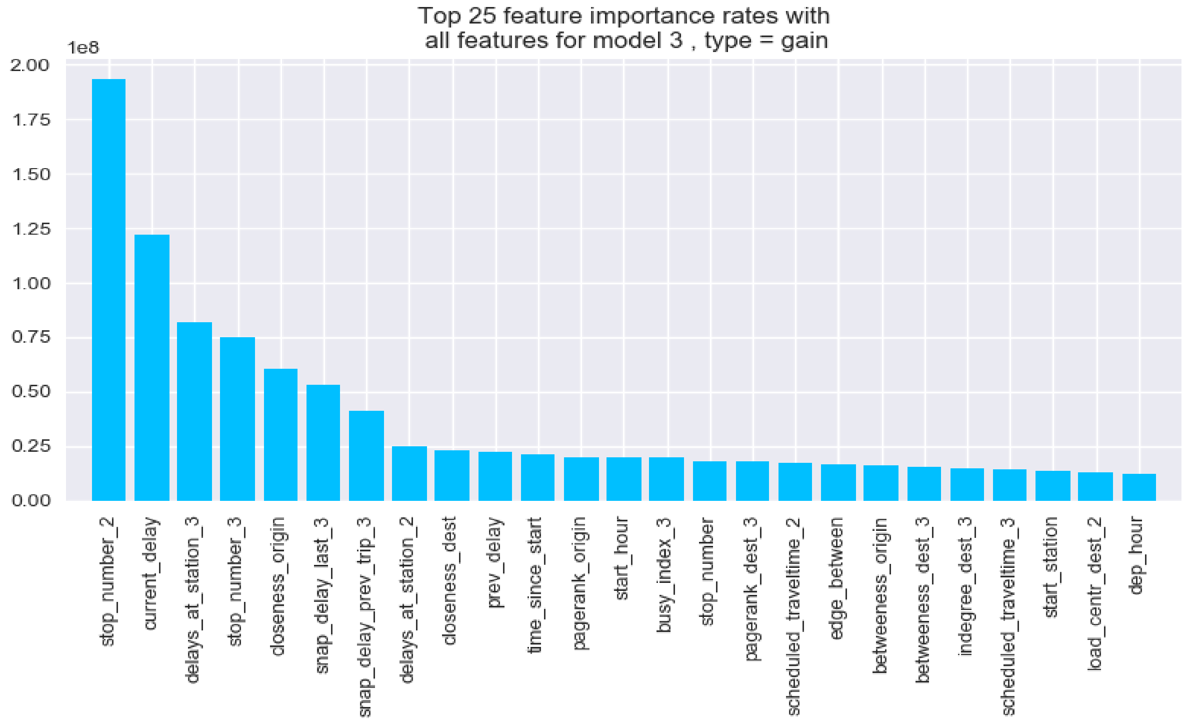


Figure 25: Top 25 feature importance scores during K-fold CV. Each feature score has been averaged over all folds

To sum up, the input features from the category delay-propagation have achieved the highest feature importance scores, especially for model $\boldsymbol{\vartheta_1}$. Further, not including this input feature group resulted in a significant drop in prediction accuracy for all models. In addition, we could see that leaving out some network-related features lead to a drop in prediction accuracy for models $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$. Station-related features derived from timetable do not visibly contribute to prediction accuracy of any model. Feature importance plots indicated that closeness-centrality was the station-related feature that has contributed most within the models of all station-related features. Trip-related features especially *stop_number*, which indicates the station index within a trip, shows growing importance for model $\boldsymbol{\vartheta_2}$ and $\boldsymbol{\vartheta_3}$, while the delay-propagation features are getting less important. Leaving out snapshot features did not have any significant impact on prediction accuracy as well. Further, their contribution within the prediction models were constant over all three models. *snap_delay_last* contributed most within this input feature group.

6.2.2 Learning curve

Computing the learning curves allows to assess how the prediction model perform differently with different amount of trainig data. In figure 26 the cross-validation score $\boldsymbol{R^2}$ on the y-axis and the training examples on the x-axis. The learning curve has been produced using model $\boldsymbol{\vartheta_1}$ using all features as after all the aim would be to assess the overall prediction power of this model. We can see that with increasing training examples the cross-validation score is getting higher. Further, the corresponding training score is getting lower, which is good because high training score would indicate overfitting the prediction model. Cross-validation score indicates stagnation after 1’500’000 training examples at a score of 0.68. This indicates that even more training samples would not lead to improved prediction results. Consequently, it represents the limits of the prediction model given the input features available.

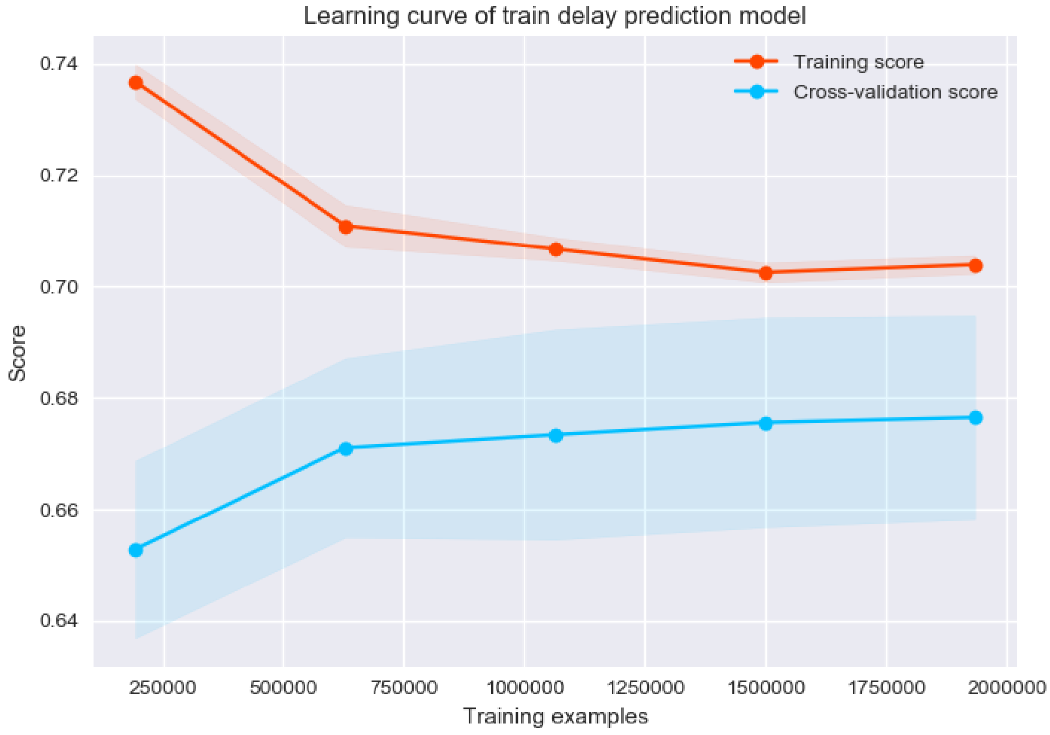


Figure 26: Learning curve showing cross-validation score measured in R^2 with increasing amount of training samples

Furthermore, one can see that training score is slightly increasing using 1’800’000 samples indicating an overfitting of the model. In order to avoid this, one could change the hyperparameter settings.

7. Discussion

The general question of this thesis was to investigate on how different input features contribute to the prediction of arrival delays. For this a train arrival delay prediction system has been built, which was used to perform different predictions using different combinations of the input feature space. In the following the results presented in chapter 6 will be discussed in regard to the research question formulated in 2.5.

Research question 1: How do different feature categories, as being used in existing literature, contribute to train arrival delay prediction?

From the results described in chapter 6 the highest prediction accuracies could be achieved using all proposed input variables, which includes network-related and snapshot variables. On the one side, it has been seen that snapshot features correlate with arrival-delays. In addition, they showed stable feature importance scores for predicting the arrival delay of three subsequent arrival stations. Therefore, input features following the snapshot principle as described in chapter 4.2.3 are senseful input variables to predict arrival delays using machine-learning algorithms. The snapshot variables can be seen as baseline-predictors for the machine-learning algorithm, that give an overall guideline of the situation between the leg-origin and the leg-destination station. On the other hand, with decreasing number of trips within a network the snapshot variables should be used carefully. As the time gap between the previous and the actual trip might be higher and therefore the encountered situation by a train-servicer could be very different. This is a problem that has been recently addressed by Sun et al. (2018) by classifying historical snapshot variables in order to detect outliers.

Network-related features are challenging to compute, as a deep understanding in the data to identify the connecting relations between different trains is needed. Further, as we could see in Oneto et al’s (2018) approach, this can lead to many interdependent models being needed, as the input table dimensions change. Nevertheless, results show that network-related features influence the prediction accuracy of a train delay prediction model positively. Especially, if the prediction task is characterized by larger time intervals between the current moment and the arrivals to be predicted. But still it is expectable that the contribution of network-related features to mirror the current traffic situation would also steadily decrease for prediction horizons lying beyond the next three stations. A distinction should be made between network-related features that capture the current traffic situation within the railway network, such as *delays_at_station* and network-related that are based on timetable information (*busy-index*). Due to the inert nature of railway networks regarding delay recovery, input variables capturing the current situation are important for train arrival delay prediction, wherefore network-related features based on scheduled plans no improvement in prediction accuracy could be detected.

Similar findings have been found by trip-related features. Here, *stop_number* is the input feature that provides best accuracy improvements when used for splitting within gradient boosted regression tress. *Stop_number* has especially contributed for the arrival delay prediction within model $\boldsymbol{\vartheta_3}$. It seems plausible that the probability of a delay increases with increasing number of stops on the trip. The reason that travel time does not contribute to arrival delay prediction might be due to the fact that effects of passenger boarding and transfers at stations are neglected. Furthermore, as the railway operator SBB assesses delays by passenger punctuality, re-scheduling and dispatching of trains might be undertaken to guarantee transfers at station. This may lead to delayed departure of train-services, as they need to wait for other trains to guarantee the transfers between lines. Consequently, waiting trains will depart late resulting to arrival delays at the following station. This scenario could explain why the contribution of *stop_number* is higher to predict train arrival delay than *scheduled_travel_time* or *time_since_start*.

The most important input feature category that could be detected was undisputable the category of delay propagation. Besides the highest loss in prediction accuracy when leaving out, the input features of this category have shown very high feature importance scores for the prediction of the direct following train arrival. As we can see in figure 23, the input feature *current_delay* resulted in large accuracy improvements when used to split decision trees within the gradient boosted regression tree model. On the other side, for predicting arrival delays beyond the following station the feature importance scores are lower, but so is the overall prediction accuracy of the model. Consequently, it cannot be stated that delay-propagation features are not important for predicting arrival delays at stations beyond the following station.

To sum up, snapshot variables are important feature to predict train arrival delay. They can provide a guideline for the expected delay, which is also reflected by high feature importance scores for model $\boldsymbol{\vartheta_1}$, see figure 23. They could be improved in combination using a collection of all past snapshot arrival delays to detect anomalies of the current snapshot delay. This would allow to assess the quality of the individual snapshot feature. Network-related features are also input variables that contribute to arrival delay prediction, especially for arrival events that lie beyond the following stations. However, one need to distinguish between network-related features that mirror the current traffic situations and network-related features based on timetables. For the latter no contribution for train arrival prediction could be found.

Research question 2: How do input features capturing topological properties between stations contribute to train arrival delay prediction?

Input features measuring the topological relations between direct connected stations and between stations and their network derived from timetable analysis did not contribute to the prediction of arrival delays. After all, the most promosing measure were betweenness- and closeness-centrality but still the relative importance within the machine learning algorithm were rather small. In addtion, leaving out the station-related input features to predict arrival delay did not resulted in a loss in prediction accuracy. Even if the same has been found for snapshot features and partly for network-related feature, it is doubtful whether station-related features contribute to arrival delay prediction as snapshot or network-related features do. This is indicated by low feature importance scores for all three models, followed by nearly no linear correlation with arrival delays. The question arises why these input features do not contribute to arrival delay prediction. One possible reason could be the approach used to model the network topologies. In figure 14 the modelling approach is illustrated, which would correspond to a simple public transport map according to Von Ferber et al. (2009). But it has been shown, that the results of centrality measures differ between the modelling approach used (Von Ferber *et al.*, 2007; Derrible and Kennedy, 2011). Consequently, using another representation as proposed by Von Ferber et al. (2009) may lead to other results. Another possible reason is that the proposed station-related features do not contribute to arrival delay prediction might be the quality of the timetable itself. Timetable scheduling is a challenging task requiring a lot of know-how and experience (Törnquist, 2006). Further, timetable quality is a key factor for the punctuality of the train services. But, if the prerequisites are in place to create a well-functioning timetable combined with the necessary operational know-how and infrastructure to execute this timetable. Then measuring topological relations within the timetable might not be a senseful strategy to engineer input features that should contribute to arrival delays prediction.

Consequently, it can be proposed that topological relations within the timetable should be put in context with other information. For example, the number of tracks available at a station or the number and position of railroad switches within the trip-leg. However, one should not solely focus on infrastructure-related properties. Passenger volumne per station could also be considered, as large passenger volumnes could cause prolonged passenger boardings (Lee, Yen and Chou, 2016). Another possibilty could be to calculate catchment areas for each station of the long-distance traffic railway network using the number of incoming regional trains and local buses. This could be an indicator of the regional importance of railway station in turn this could be an indicator for the passenger volumne at the station. Thus, from a railway operators perspective, which uses passenger punctuality measures instead of effective arrival delays for service-quality assessments. All these information could be useful to weight nodes within the topological network and thus improve the centality measures in order to find underlying relations with train arrival delay.

Research question 3: How does the prediction model perform differently with different amount of training data?

Considering the learning curve presented in figure 26 it can be concluded that increasing the number of training samples leads to a higher $\boldsymbol{R^2}$ score. Consequently, the model can better explain the proportion of variance of the dependent variable, which is satisfying. On the other hand it is clearly visible that a threshold can be expected around $\boldsymbol{R^2}$ = 0.68. In addition, the slight increase in training score resulting with nearly 2’000’000 samples is an indicator that the model starts to overfit with this amount of data. From this, it can be concluded that the amount of data was not the factor limiting the prediction accuracy of the proposed model. Nevertheless, this could be different using other machine-learning algorithms. In order to achieve better scores using the prediction model as proposed here, one should focus on the process of feature engineering. For example, engineering better network-related or snapshot features as discussed in the previous research question. Satisfying is the fact that the prediciton model is able to achieve good results with little less than 1’000’000 observations which would correspond to train-describer data of around 2.5 to 3 months in the case of the long-distance traffic in Switzerland.

Final thoughts and outlook

The focus of this thesis lied on the evaluation of different input features to predict train arrival delays. For this, a prediction system has been built using machine-learining techniques. All in all, it is difficult to compare the proposed train delay prediction system with other existing prediction systems. Nevertheless, the achieved results were satisfying as given a set of input features the arrival delay for the three subsequent stations can be predicted with an average root mean square error of 107, 118 and 121 seconds. Further, different feature categories could be identified as important variables to predict arrival delays for different prediction scenarios, such as short or long prediction horizons. Moreover, it could be found that input features measuring topological properties predefined in the timetable do not contribute to arrival delay prediction. It can be proposed that these input features should be enriched with other railway operation relevant data to close the gap between the planned topological timetable relations and the in reality executed railway operations.

8. References:

As, M. and Mine, T. (2018) ‘Dynamic Bus Travel Time Prediction Using an ANN-based Model’, in *IMCOM ’18 Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*, p. 20. doi: 10.1145/3164541.3164630.

Assunção, M. D. *et al.* (2015) ‘Big Data computing and clouds: Trends and future directions’, *Journal of Parallel and Distributed Computing*. Elsevier Inc., 79–80, pp. 3–15. doi: 10.1016/j.jpdc.2014.08.003.

Batley, R., Dargay, J. and Wardman, M. (2011) ‘The impact of lateness and reliability on passenger rail demand’, *Transportation Research Part E: Logistics and Transportation Review*. Pergamon, 47(1), pp. 61–72. doi: 10.1016/J.TRE.2010.07.004.

Berger, A. *et al.* (2011) ‘Stochastic Delay Prediction in Large Train Networks’, *DROPS-IDN/3270*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik. doi: 10.4230/OASICS.ATMOS.2011.100.

Brandes, U. (2008) ‘On variants of shortest-path betweenness centrality and their generic computation’, *Social Networks*, 30(2), pp. 136–145. doi: 10.1016/j.socnet.2007.11.001.

Van Bruggen, R. (2015) *Bruggen Blog: Loading General Transport Feed Spec (GTFS) files into Neo4j - part 1/2*. Available at: http://blog.bruggen.com/2015/11/loading-general-transport-feed-spec.html (Accessed: 26 September 2019).

Bundesamt für Statistik (2016) *Pendlermobilität in der Schweiz 2016, Mit einer Vertiefung zu den Pendlerströmen zwischen den Gemeinden*. Available at: https://www.bfs.admin.ch/bfs/de/home/statistiken/mobilitaet-verkehr/personenverkehr/pendlermobilitaet.assetdetail.5827316.html.

Bundesamt für Verkehr BAV (2017) *Wegleitung: Grundsätze und Kriterien für den Fernverkehr Version 2.0*. Available at: https://www.bav.admin.ch/bav/de/home/das-bav/aufgaben-des-amtes/finanzierung/finanzierung-verkehr/personenverkehr/fernverkehr-fv.html.

Čelan, M. and Lep, M. (2017) ‘Bus arrival time prediction based on network model’, *Procedia Computer Science*. Elsevier B.V., 113, pp. 138–145. doi: 10.1016/j.procs.2017.08.331.

Chen, M. *et al.* (2004) ‘A Dynamic Bus-Arrival Time Prediction Model Based on APC Data’, *Computer-Aided Civil and Infrastructure Engineering*. John Wiley & Sons, Ltd (10.1111), 19(5), pp. 364–376. doi: 10.1111/j.1467-8667.2004.00363.x.

Chen, T. and Guestrin, C. (no date) ‘XGBoost: A Scalable Tree Boosting System’. doi: 10.1145/2939672.2939785.

Derrible, S. (2012) ‘Network Centrality of Metro Systems’, *PLoS ONE*. Edited by P. Holme. Public Library of Science, 7(7), p. e40575. doi: 10.1371/journal.pone.0040575.

Derrible, S. and Kennedy, C. (2011) ‘Applications of graph theory and network science to transit network design’, *Transport Reviews*, 31(4), pp. 495–519. doi: 10.1080/01441647.2010.543709.

Dey, A. (2016) ‘Machine Learning Algorithms: A Review’, *International Journal of Computer Science and Information Technologies*, 7(3), pp. 1174–1179. Available at: www.ijcsit.com (Accessed: 21 January 2019).

Dotoli, M. *et al.* (2017) ‘A Decision Support System for Optimizing Operations at Intermodal Railroad Terminals’, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(3), pp. 487–501. doi: 10.1109/TSMC.2015.2506540.

Elith, J., Leathwick, J. R. and Hastie, T. (2008) ‘A working guide to boosted regression trees’, *Journal of Animal Ecology*. John Wiley & Sons, Ltd (10.1111), 77(4), pp. 802–813. doi: 10.1111/j.1365-2656.2008.01390.x.

de Fabris, S., Longo, G. and Medeossi, G. (2008) ‘Automated analysis of train event recorder data to improve micro-simulation models’, in *WIT Transactions on The Built Environment*. WIT Press, pp. 575–583. doi: 10.2495/CR080561.

Fayyaz S., S. K., Liu, X. C. and Zhang, G. (2017) ‘An efficient General Transit Feed Specification (GTFS) enabled algorithm for dynamic transit accessibility analysis’, *PLoS ONE*, 12(10), pp. 1–22. doi: 10.1371/journal.pone.0185333.

Von Ferber, C. *et al.* (2007) ‘Network harness: Metropolis public transport’, *Physica A*, 380, pp. 585–591. doi: 10.1016/j.physa.2007.02.101.

Von Ferber, C. *et al.* (2009) ‘Public transport networks: empirical analysis and modeling’, *Eur. Phys. J. B*, 68, pp. 261–275. doi: 10.1140/epjb/e2009-00090-x.

Fortin, P., Morency, C. and Trépanier, M. (2016) ‘Innovative GTFS Data Application for Transit Network Analysis Using a Graph-Oriented Method’, *Journal of Public Transportation*, 19(4), pp. 18–37. doi: 10.5038/2375-0901.19.4.2.

Fosso Wamba, S. *et al.* (2015) ‘How “big data” can make big impact: Findings from a systematic review and a longitudinal case study’, *International Journal of Production Economics*, 165(January), pp. 234–246. doi: 10.1016/j.ijpe.2014.12.031.

Freeman, L. C. (1979) ‘Centrality in social networks conceptual clarification’, *Social Networks*, 1(3), pp. 215–239. doi: 10.1016/0378-8733(78)90021-7.

Fumeo, E., Oneto, L. and Anguita, D. (2015) ‘Condition Based Maintenance in Railway Transportation Systems Based on Big Data Streaming Analysis’, *Procedia Computer Science*. Elsevier, 53, pp. 437–446. doi: 10.1016/J.PROCS.2015.07.321.

Gal, A. *et al.* (2017) ‘Traveling time prediction in scheduled transportation with journey segments’, *Information Systems*. Elsevier, 64, pp. 266–280. doi: 10.1016/j.is.2015.12.001.

Ghofrani, F. *et al.* (2018) ‘Recent applications of big data analytics in railway transportation systems: A survey’, *Transportation Research Part C: Emerging Technologies*, 90, pp. 226–246. doi: 10.1016/j.trc.2018.03.010.

Google (2015) *General Transit Feed Specification Reference*. Available at: https://developers.google.com/transit/gtfs/reference/ (Accessed: 20 May 2018).

Goto, H. (2014) ‘Introduction to max-plus algebra’, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation, ISSAC*. New York, New York, USA: ACM Press, pp. 21–22. doi: 10.1145/2608628.2627496.

Goverde, R. M. P. (2010) ‘A delay propagation algorithm for large-scale railway traffic networks’, *Transportation Research Part C: Emerging Technologies*. Elsevier Ltd, 18(3), pp. 269–287. doi: 10.1016/j.trc.2010.01.002.

Goverde, R. M. P. and Hansen, I. A. (2000) ‘TNV-prepare: Analysis of Dutch railway operations based on train detection data’, *Advances in Transport*, 7, pp. 779–788.

Gurmu, Z. and Fan, W. (2014) ‘Artificial Neural Network Travel Time Prediction Model for Buses Using Only GPS Data’, *Journal of Public Transportation*, 17(2), pp. 45–65. doi: 10.5038/2375-0901.17.2.3.

Hadas, Y. *et al.* (2014) ‘Public transport systems’ connectivity: Spatiotemporal analysis and failure detection’, *Transportation Research Procedia*. Elsevier B.V., 3(July), pp. 309–318. doi: 10.1016/j.trpro.2014.10.011.

Hansen, I. A., Goverde, R. M. P. and Van Der Meer, D. J. (2010) ‘Online train delay recognition and running time prediction’, in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1783–1788. doi: 10.1109/ITSC.2010.5625081.

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *Springer Series in Statistics The Elements of Statistical Learning*. 2nd edn, *The Mathematical Intelligencer*. 2nd edn. Springer. doi: 10.1007/b94608.

Huang, G., Zhu, Q. and Siew, C. (2006) ‘Extreme Learning Machine : Theory and Applications Extreme learning machine : Theory and applications’, *Neurocomputing*, 70(1–3), pp. 489–501. doi: 10.1016/j.neucom.2005.12.126.

Huang, H. *et al.* (2018) ‘Multimodal Route Planning With Public Transport and Carpooling’, *IEEE Transactions on Intelligent Transportation Systems*. IEEE, PP, pp. 1–13. doi: 10.1109/TITS.2018.2876570.

Hyndman, R. J. and Athanasopoulos, G. (2018) *Forecasting: Principles and Practice*. 2nd edn, *OTexts*. 2nd edn. Melbourne, Australia: OTexts. doi: 10.1017/9781316451038.010.

Jain, A. (2015) *Complete Guide to Parameter Tuning in XGBoost (with codes in Python)*. Available at: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/ (Accessed: 28 September 2019).

James, G. *et al.* (2013) *An Introduction to Statistical Learning, Journal of Chemical Information and Modeling*. New York: Springer. doi: 10.1017/CBO9781107415324.004.

Jeong, R. and Rilett, R. (2004) ‘Bus arrival time prediction using artificial neural network model’, in *Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749)*. IEEE, pp. 988–993. doi: 10.1109/ITSC.2004.1399041.

Kaleko, D. (2017) *From Neutrinos to Data Science*. Available at: http://blog.davidkaleko.com/feature-engineering-cyclical-features.html (Accessed: 28 September 2019).

Kecman, P. and Goverde, R. M. P. (2015) ‘Online data-driven adaptive prediction of train event times’, *IEEE Transactions on Intelligent Transportation Systems*, 16(1), pp. 465–474. doi: 10.1109/TITS.2014.2347136.

Kujala, R. *et al.* (2018) ‘Travel times and transfers in public transport: Comprehensive accessibility analysis based on Pareto-optimal journeys’, *Computers, Environment and Urban Systems*, 67, pp. 41–54. doi: 10.1016/j.compenvurbsys.2017.08.012.

Lee, W. H., Yen, L. H. and Chou, C. M. (2016) ‘A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services’, *Transportation Research Part C: Emerging Technologies*.

Pergamon, 73, pp. 49–64. doi: 10.1016/j.trc.2016.10.009.

Lessan, J., Fu, L. and Wen, C. (2019) ‘A hybrid Bayesian network model for predicting delays in train operations’, *Computers & Industrial Engineering*. Elsevier, 127, pp. 1214–1222. doi: 10.1016/j.cie.2018.03.017.

Li, H. *et al.* (2014) ‘Improving rail network velocity: A machine learning approach to predictive maintenance’, *Transportation Research Part C: Emerging Technologies*. Pergamon, 45, pp. 17–26. doi: 10.1016/J.TRC.2014.04.013.

Liang, J., Martin, U. and Cui, Y. (2017) ‘Increasing performance of railway systems by exploitation of the relationship between capacity and operation quality’, *Journal of Rail Transport Planning and Management*. Elsevier Ltd, 7(3), pp. 127–140. doi: 10.1016/j.jrtpm.2017.08.002.

Lu, L. and Zhang, M. (2013) ‘Edge Betweenness Centrality’, in *Encyclopedia of Systems Biology*. Springer New York, pp. 647–648. doi: 10.1007/978-1-4419-9863-7_874.

Marković, N. *et al.* (2015) ‘Analyzing passenger train arrival delays with support vector regression’, *Transportation Research Part C: Emerging Technologies*, 56, pp. 251–262. doi: 10.1016/j.trc.2015.04.004.

Marsden, P. V. (2015) ‘Network Centrality, Measures of’, in *International Encyclopedia of the Social & Behavioral Sciences: Second Edition*. Elsevier Inc., pp. 532–539. doi: 10.1016/B978-0-08-097086-8.43115-6.

Meiner, H. (1991) ‘Die Entstehung des Taktfahrplans Schweiz Die Entstehung des Taktfahrplans Schweiz’, *Schweizer Ingenieur und Architekt*, 109(25), pp. 607–611.

Milinković, S. *et al.* (2013) ‘A fuzzy Petri net model to estimate train delays’, *Simulation Modelling Practice and Theory*, 33, pp. 144–157. doi: 10.1016/j.simpat.2012.12.005.

Moreira-Matias, L. *et al.* (2015) ‘Improving Mass Transit Operations by Using AVL-Based Systems: A Survey’, *IEEE Transactions on Intelligent Transportation Systems*. doi: 10.1109/TITS.2014.2376772.

Neilson, A. *et al.* (2019) ‘Systematic Review of the Literature on Big Data in the Transportation Domain: Concepts and Applications’, *Big Data Research*. Elsevier Inc., 1, pp. 1–10. doi: 10.1016/j.bdr.2019.03.001.

Newman, M. E. J. (2001) ‘Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality’, *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 64(1), p. 7. doi: 10.1103/PhysRevE.64.016132.

Oneto, L. *et al.* (2016) ‘Advanced analytics for train delay prediction systems by including exogenous weather data’, *Proceedings - 3rd IEEE International Conference on Data Science and Advanced Analytics, DSAA 2016*, pp. 458–467. doi: 10.1109/DSAA.2016.57.

Oneto, L. *et al.* (2017) ‘Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout’, *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(10), pp. 2754–2767. doi: 10.1109/TSMC.2017.2693209.

Oneto, L. *et al.* (2018) ‘Train Delay Prediction Systems: A Big Data Analytics Perspective’, *Big Data Research*. Elsevier Inc., 11, pp. 54–64. doi: 10.1016/j.bdr.2017.05.002.

Oruganti, A. *et al.* (2016) ‘DelayRadar: A multivariate predictive model for transit systems’, *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, pp. 1799–1806. doi:

10.1109/BigData.2016.7840797.

Page, L. *et al.* (1998) ‘The PageRank Citation Ranking: Bringing Order to the Web’, *World Wide Web Internet And Web Information Systems*, 54(1999–66), pp. 1–17. doi: 10.1.1.31.1768.

Pandey, V. and Kemper, A. (2016) ‘Big Geospatial Data Exploration’, *Digital Mobility Plattforms and Ecosystems - State of the Art Report*, (July), pp. 212–218.

Peters, J. *et al.* (2005) ‘Prediction of delays in public transportation using neural networks’, in *Proceedings - International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2005 and International Conference on Intelligent Agents, Web Technologies and Internet*. IEEE, pp. 92–97. doi: 10.1109/CIMCA.2005.1631451.

Petersen, T. (2016) ‘Watching the Swiss: A network approach to rural and exurban public transport’, *Transport Policy*. Elsevier, 52, pp. 175–185. doi: 10.1016/j.tranpol.2016.07.012.

Pongnumkul, S. *et al.* (2014) ‘Improving arrival time prediction of Thailand’s passenger trains using historical travel times’, *2014 11th Int. Joint Conf. on Computer Science and Software Engineering: ‘Human Factors in Computer Science and Software Engineering’ - e-Science and High Performance Computing: eHPC, JCSSE 2014*. IEEE, pp. 307–312. doi: 10.1109/JCSSE.2014.6841886.

Psaltoglou, A. and Calle, E. (2018) ‘Enhanced connectivity index – A new measure for identifying critical points in urban public transportation networks’, *International Journal of Critical Infrastructure Protection*, 21, pp. 22–32. doi: 10.1016/j.ijcip.2018.02.003.

Rao, H. *et al.* (2019) ‘Feature selection based on artificial bee colony and gradient boosting decision tree’, *Applied Soft Computing Journal*. Elsevier B.V., 74, pp. 634–642. doi: 10.1016/j.asoc.2018.10.036.

Sameni, M. K., Landex, A. and Preston, J. (2011) ‘Developing the UIC 406 method for capacity analysis’, in *4th International Seminar on Railway Operations Research*, pp. 1–19. Available at: http://forskningsbasen.deff.dk/Share.external?sp=S8e45a277-56b5-46b2-b6dc-8aad418c025f&sp=Sdtu (Accessed: 27 September 2019).

SBB (2018) *Pünktlichkeit unter der Lupe | SBB News*. Available at: https://news.sbb.ch/artikel/76242/puenktlichkeit-unter-der-lupe (Accessed: 29 September 2019).

Schweizerische Bundesbahnen (2017) ‘Unser Versprechen für die Schweiz und ihre Regionen’. doi: 10.1016/j.desal.2005.12.027.

Senderovich, A. *et al.* (2014) ‘Queue Mining – Predicting Delays in Service Processes’, in *Jarke M. et al. (eds) Advanced Information Systems Engineering. CAiSE 2014. Lecture Notes in Computer Science, vol 8484.* Springer, Cham, pp. 42–57. doi: 10.1007/978-3-319-07881-6_4.

Shafabakhsh, G. A., Famili, A. and Bahadori, M. S. (2017) ‘GIS-based spatial analysis of urban traffic accidents: Case study in Mashhad, Iran’, *Journal of Traffic and Transportation Engineering (English Edition)*. Elsevier Ltd, 4(3), pp. 290–299. doi: 10.1016/j.jtte.2017.05.005.

Shalaby, A. and Farhan, A. (2003) ‘Bus Travel Time Prediction Model for Dynamic Operations Control and Passenger Information Systems’, in *The 82nd Annual Meeting of the Transportation Research Board*. Available at:

http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=07E3B0441892523097AE76E7E13FE581?doi=10.1.1.468.1325&rep=rep1&type=pdf (Accessed: 3 April 2019).

Sun, F. *et al.* (2018) ‘Transit-hub: a smart public transportation decision support system with multi-timescale analytical services’, *Cluster Computing*, pp. 1–16. doi: 10.1007/s10586-018-1708-z.

Sun, Y., Wong, A. K. C. and Kamel, M. S. (2009) ‘Classification of imbalanced data: A review’, *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4), pp. 687–719. doi: 10.1142/S0218001409007326.

To, W. M. (2015) ‘Centrality of an Urban Rail System’, *Urban Rail Transit*. Springer Berlin Heidelberg, 1(4), pp. 249–256. doi: 10.1007/s40864-016-0031-3.

Törnquist, J. (2006) ‘Computer-based decision support for railway traffic scheduling and dispatching: A review of models and algorithms’, in *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS’05) 2*, p. 23p. Available at: http://drops.dagstuhl.de/opus/volltexte/2006/659 (Accessed: 14 May 2019).

Tu, Y. (2013) ‘Centrality characteristics analysis of urban rail network’, in *IEEE ICIRT 2013 - Proceedings: IEEE International Conference on Intelligent Rail Transportation*. IEEE, pp. 285–290. doi: 10.1109/ICIRT.2013.6696309.

Wang, R. and Work, D. B. (2015) ‘Data Driven Approaches for Passenger Train Delay Estimation’, in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*. IEEE, pp. 535–540. doi: 10.1109/ITSC.2015.94.

XGBoost, D. (2019) ‘xgboost Release 0.82’. Available at: https://buildmedia.readthedocs.org/media/pdf/xgboost/release_0.82/xgboost.pdf.

Xie, Z. and Yan, J. (2013) ‘Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: An integrated approach’, *Journal of Transport Geography*. Elsevier Ltd, 31, pp. 64–71. doi: 10.1016/j.jtrangeo.2013.05.009.

Yaghini, M., Khoshraftar, M. M. and Seyedabadi, M. (2013) ‘Railway passenger train delay prediction via neural network model’, *Journal of Advanced Transportation*. John Wiley & Sons, Ltd, 47(3), pp. 355–368. doi: 10.1002/atr.193.

Yamaguchi, T., As, M. and Mine, T. (2019) ‘Prediction of Bus Delay over Intervals on Various Kinds of Routes Using Bus Probe Data’, *2018 IEEE/ACM 5th International Conference on Big Data Computing Applications and Technologies (BDCAT)*. IEEE, pp. 97–106. doi: 10.1109/bdcat.2018.00020.

Yuan, J., Goverde, R. M. P. and Hansen, I. A. (2006) ‘Evaluating stochastic train process time distribution models on the basis of empirical detection data’, in *WIT Transactions on the Built Environment*, pp. 631–640. doi: 10.2495/CR060621.

Zhang, J. *et al.* (2013) ‘Networked characteristics of the urban rail transit networks’, *Physica A: Statistical Mechanics and its Applications*. North-Holland, 392(6), pp. 1538–1546. doi: 10.1016/J.PHYSA.2012.11.036.

Zhong, C. *et al.* (2014) ‘Detecting the dynamics of urban structure through spatial network analysis’, *International Journal of Geographical Information Science*, 28(11), pp. 2178–2199. doi: 10.1080/13658816.2014.914521.

Zhong, C. *et al.* (2015) ‘Measuring variability of mobility patterns from multiday smart-card data’, *Journal of Computational Science*. doi: 10.1016/j.jocs.2015.04.021.

Zychowski, A., Junosza-Szaniawski, K. and Kosicki, A. (2018) ‘Travel time prediction for trams in Warsaw’, *Advances in Intelligent Systems and Computing*, 578(688380), pp. 53–62. doi: 10.1007/978-3-319-59162-9_6.

Personal declaration

I hereby declare that the submitted thesis is the result of my own, independent work. All external sources are explicitly acknowledged in this thesis

O.Niklaus

September 30rd, 2019

Olivier Niklaus