Master’s thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Caroline Skjøren
Anders Sørskår Larsen

Predictions for Railway Traveller Information

Developing a new prediction model, and comparing it to Bane NOR's existing model

Master’s thesis in Engineering and ICT
Supervisor: Nils Olsson
June 2023

NTNU
Norwegian University of
Science and Technology

Caroline Skjøren
Anders Sørskår Larsen

Predictions for Railway Traveller Information

Developing a new prediction model, and comparing it to Bane NOR's existing model

Master's thesis in Engineering and ICT
Supervisor: Nils Olsson
June 2023

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Caroline Skjøren
Anders Sørskår Larsen

Predictions for Railway Traveller Information

Developing a new prediction model, and comparing it to Bane NOR's existing model

Master's thesis in Project and Quality Management
Supervisor: Nils O.E. Olsson
June 2023

Norwegian University of Science and Technology
Faculty of Natural Sciences
Department of Mechanical and Industrial Engineering

NTNU

PREFACE

This paper is a Master‘s thesis in Project and Quality Management at the Department of Mechanical and Industrial Engineering at Norwegian University of Science and Technology (NTNU) in Trondheim, carried out in the spring of 2023. The master is a part of the five-year master’s study program Engineering and ICT where both students have chosen the main profile ICT and Machine Technology. This paper is based on both authors preliminary specialization projects, written during the autumn semester of 2022. The supervisor, Nils Olsson, have been supervising both the individual specialization projects, and this master thesis.

The project *"Predictions for railway traveler information"* was chosen due to both author’s interest in the field of Artificial Intelligence and Machine Learning. The prospect of applying theoretical machine learning knowledge to a complex real-world problem really caught our interest.

ACKNOWLEDGEMENTS

We want to give a thanks to the people involved. Our main supervisor Professor Nils Olsson, from the Department of Mechanical and Industrial Engineering, have supervised both our specialisation projects, in addition to this master's thesis. Nils have been very helpful with regular and pleasant meetings, giving answers and feedback whenever needed. His expertise and patience have been invaluable and have played a crucial role in the success of this thesis.

We would also like to thank Postdoctoral Pranjal Mandhaniya for discussing the master's thesis' results with us and providing valuable feedback.

Senior researcher Andreas Dypvik Landmark has been helping the construction of and advising throughout the master's thesis.

We are also grateful to our connections in Bane NOR. Svein Eriksen, Kjetil Hovden and Tone Norløff have been answering questions, collecting and delivering the historical train data.

Emma Sofie Skarstein and Kenneth Aase, both PhD-students and teacher assistants in TMA4268, have helped improve the overall research quality and internal validity. Research findings, interpretations and conclusions have been shared with them in order to obtain valuable feedback and criticism.

We would also like to give our friends and family warm thanks for their support throughout five years in Trondheim.

ABSTRACT

The railway plays an essential role in terms of people´s daily transportation. Accurate real-time train delay prediction is essential for efficient railway traffic planning and management as well as for delivering adequate passenger service quality. However, predicting train delays is difficult due to the dynamics and uncertainty of the evolution of the delay.

This master's thesis delves into the realm of predictive modelling in railway operations, exploring the effectiveness of Long Short-Term Memory (LSTM) networks, a deep learning technique, in predicting everyday train delays. It offers a comparative analysis of the LSTM model against Bane NOR's current train departure prediction model. Four key research questions guided the investigation, focusing on the utility of LSTM in delay prediction, the impact of various train and weather features on the model's accuracy, and the comparative performance of the LSTM and Bane NOR's models.

The study revealed that the LSTM model, even when devoid of weather features, outperformed Bane NOR's model in predicting train departures. This suggests that the LSTM model's ability to capture and learn from historical delay patterns may be a significant contributor to its predictive accuracy. However, the inclusion of weather data did not improve the LSTM model's performance as anticipated, potentially due to limitations such as missing weather data and biases introduced during the data imputation process. Despite these challenges, the LSTM model's superior performance highlights the potential of deep learning techniques in enhancing the precision and reliability of delay predictions in railway operations.

These findings extend the burgeoning discourse on the application of machine learning in transportation modelling. In essence, this thesis provides evidence that LSTM models could serve as a valuable tool in railway operations, promising more accurate train delay predictions and, in turn, contributing to improved operational efficiency and customer satisfaction.

SAMMENDRAG

Jernbanen har alltid vært en viktig samfunnsaktør for hverdagslig transport. For å kunne opprettholde en effektiv jernbanetrafikkplanlegging og styring, så er nøyaktig sanntids tog forsinkelses prediksjoner essensielt. Med det sagt, predikering av tog forsinkelser er en vanskelig oppgave med tanke på kompleksiteten av dynamikken og uforutsigbarheten av en forsinkelsesutvikling.

Denne masteroppgaven fordyper seg i området for prediktiv modellering av forsinkelser på jernbanen. Studiet utvikler en prediksjonsmodell basert på det kunstig nevrale nettverket Long Short-Term Memory (LSTM), som er en dyp læring teknikk for å forutsi daglige togforsinkelser. Prediksjonsmodellens effektivitet blir videre utforsket, analysert og sammenliknet med den norske operative prediksjonsmodell, som er i bruk av Bane NOR. Med fokus på LSTM prediksjonsmodellens nytteverdi, modellens nøyaktighets påvirkning av ulike tog- og værparametere, og den komparative ytelse og prestasjon av LSTM og Bane NOR´s prediksjonsmodeller.

Studiet viser lovende estimater for togavganger av LSTM modellen. Dette gjenspeiler LSTM modellen sin egenskap til å fange opp og lære mønstre i jernbanen basert på historisk data, og viser hvordan egenskapen kan benyttes til presisering i prediktiv modellering. Resultatene til LSTM modellen er mer nøyaktig og realistisk ved sammenlikning av Bane NOR´s modell. Videre viser studiet hvordan inkludering av værdata ikke forbedret LSTM-modellens prediksjoner. Det kan være en konsekvens av studiets egne begrensninger og dataprosesserings prosess, sånn som manglende værdata, begrenset tidsperiode og skjevheter i dataen som et biprodukt av imputerings prosessen. LSTM modellen har vist høy evne gjennom dyplæringsteknikker med implementerte beskrivende jernbanefaktorer til å forbedre presisjonen og påliteligheten til forsinkelsesforutsigelser i jernbanedrift.

Masteroppgavens funn viser mulighetsrommet for å utvide bruken av maskinlæringsteknikker i transportsektoren. Oppgaven illustrerer hvordan LSTM-modeller kan være et verdifullt verktøy i jernbanedrift. Med nøyaktige togforsinkelsesprediksjoner kan modellen på sin side bidra til forbedret driftseffektivitet og kundetilfredshet.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

List of all abbreviations in alphabetic order:

- **API** Application Programming Interface
- **ATD** Actual Time of Departure
- **DNN** Deep Neural Network
- **ETD** Estimated Time of Departure
- **EV** External Variables
- **LSTM** Long Short-Term Memory Networks
- **MAE** Mean Absolute Error
- **MET** Meteorologisk Institutt (Meteorological Institute)
- **NN** Neural Network
- **NTNU** Norwegian University of Science and Technology
- **RMSE** Root Mean Square Error
- **RNN** Recurrent Neural Network
- **RQ** Research Question
- **SIRI-ET** SIRI Estimated Timetable
- **SIRI-SX** SIRI Situation Exchange
- **SIRI-VM** SIRI Vehicle Monitoring
- **STD** Scheduled Time of Departure
- **STV** Spatio Temporal Variables
- **TIOS** Trafikkinformasjons- og oppfølgingssystem
- **XML** Extensible Markup Language

CHAPTER

ONE

INTRODUCTION

The railway plays an essential role in terms of people´s daily transportation. Being reliable and adhering to the schedule with little deviation is one of the most crucial service level requirements for railway companies. Accurate real-time train delay prediction is essential for efficient railway traffic planning and management as well as for delivering adequate passenger service quality.

The forecasting has a well-established role in all levels of planning, control and management of railway traffic. Effective route planning, traffic management, rescheduling, and passenger information all depend on accurate forecast of train movements in time and space (H. Wang et al. 2013). The information is essential to meet the goal of providing reliable train services to transport passengers or goods, allowing them to take proactive actions to lessen the impact of train delays. Traffic controllers predict arrival times of the trains within their region to manage the feasibility of timetable realisation. Transport controllers, working for railway operating firms, evaluate the feasibility of planned passenger transfers as well as rolling-stock and staff rotation schemes (Kauppi et al. 2017).

Railway operations depends on timetables, which are described as "a regular vehicle movement between two or more points at pre-scheduled times, transporting passengers or cargo" (Spanninger et al. 2022). Due to every path passing via multiple locations and each participant's dependence on and influence over one another, the interconnected system is very complicated. Further, railway networks often operate at or near full capacity due to an increasing demand. The complexity of railway operations exacerbated by delays that might distribute in temporal and spatial directions, causing delay propagation on the line or even on the network. Train delays are inconvenient for railway operators and their customers as they reduce train service quality where customers miss their transfers, trains are cancelled, and travel time is increased. These delays are inevitable as many uncertainties caused by for instance incidents such as passenger crowds, infrastructure failures, or extreme weather, which make the timetable and perhaps the resource obligations unachievable.

Delay propagation is a major contributor to deviations in railway systems (Yuan et al. 2002, P. Huang, Zhongcan Li et al. 2021). Making rational decisions is necessary to prevent delay propagation, and valid estimations of times of arrival and departure are seen as fundamental components of these decisions. However, anticipating train events is a challenging task due to the uncertainty and unpredictability of process time in railway traffic, dealing with the impacts of mishaps, and bringing trains back into normal service. The process is challenging because it must take into account society expectations and trends, industrial conditions, political, economic, and psychological issues (Profillidis 2006).

Due to increasing data quantities and availability, researchers are now able to study delays and their causes at a very comprehensive level. The likelihood that the solutions work well is higher than ever as techniques and approaches progress and train demand rises. This is valid not just for ordinary operations but also in the event of unanticipated events or actual realizations of predicted parameters (Lusby et al. 2018a).

Recent advancements in artificial intelligence (AI) offer new possibilities for addressing the limitations of traditional models. Deep learning approaches, such as Long Short-Term Memory (LSTM) models, can learn complex patterns and relationships within large, high-dimensional data sets. These abilities make these deep learning models highly suitable for a wide range of applications spanning from energy consumption forecasting to natural language processing. In contrast to traditional, static models, used for train delay prediction, deep learning models can include large amounts of features as a basis for future predictions. This is beneficial as there are potentially many factors contributing to train delays.

In this thesis, the primary focus is on predicting small, everyday delays rather than large, unpredictable delays caused by unforeseen events. These smaller delays are more commonly caused by delay accumulation along the railway line, and are crucial for maintaining high service quality and customer satisfaction.

By developing a deep learning model tailored to predict these more frequent, smaller delays, this paper aim to provide more accurate and actionable predictions for railway operators. Consequently, this will improve the real-time decision-making and overall service reliability on the railway. This approach ensures that the thesis discusses on the most relevant and prevalent types of delays, allowing for more practical and effective solutions to be proposed and evaluated.

This thesis explores the use of deep learning, specifically LSTM models, for train delay predictions. An LSTM model is a type of recurrent neural networks (RNN), capable of capturing long-term dependencies and patterns in sequential data. In this particular use-case, these patterns could translate to delay propagation and spatio-temporal dependencies along the railway line. The thesis discuss the challenges of predicting train delays, including the variety of factors, or *features*, that can contribute to delays and the variability of delay patterns over time. Further, it examines the use of different features through various feature engineering techniques and explore the incorporation of weather data as an external feature to improve predictive accuracy. Finally, the LSTM model will be compared to Bane NOR's existing prediction model.

The research questions of this thesis, listed below, are designed to explore the various facets of train delay prediction, using deep learning techniques as a primary tool. These questions form the backbone of this thesis, driving the direction of research and analysis, and aligning them with the central objective – improving the prediction of small, everyday train delays. Each research question corresponds to the challenges identified in the field of train delay prediction and aims to shed light on the potential solutions offered by deep learning techniques, primarily through the application of LSTM models.

RQ 1 How can deep learning techniques, specifically LSTM models, be used to predict small, everyday train delays?

RQ 2 How do various types of train related features contribute to an LSTM prediction model's accuracy and performance?

RQ 3 How does the incorporation of weather data, as an external feature, affect the predictive accuracy of LSTM models for train delay predictions?

RQ 4 How does the performance of the LSTM model compare to Bane NOR's existing prediction model?

This master's thesis is organised into eight chapters, each discussing different aspects of the study. The introduction, chapter 1, sets the context for the research. Chapter 2 explores a wide array of theoretical concepts related to railway networks, AI, and data usage in the railway industry, with a specific focus on the Norwegian context. In chapter 3, a review of related work is provided, including train delay prediction methods, and the role of weather conditions.

Chapter 4 presents the rail and weather data used. Chapter 5 describe the methodology used in this research, from the literature review method to feature engineering, analysis, prediction methodology, and tool selection.

Chapter 6 offers the results from the prediction model and Bane NOR´s Model, providing an in-depth comparison of both models' performance. Chapter 7 delves into a detailed discussion of the results, evaluating the models, and acknowledging limitations.

The research concludes in chapter 8 with a summary of the findings, followed by suggestions for future work.

CHAPTER

TWO

THEORY

Accurate real-time train delay prediction is crucial for the proactive and anticipatory regulation of present railway traffic control and providing railway traveller information (K. Tiong et al. 2022). The need for effective solutions is greater than ever as techniques and strategies evolve and rail usage increases. By successfully modifying the train timetable, the uncertainty surrounding passenger departure times might be reduced. The early stages of the study involved a thorough literature review, and this chapter contains the theoretical aspect. The theoretical underpinnings needed to comprehend the complexity of the railway network are first covered. Further, this chapter discusses the fundamentals required to resolve the problem statement in an efficient and effective manner. Additionally, this chapter will provide a theoretical foundation of AI and machine learning, which are essential components for developing accurate and robust train delay prediction models.

2.1 Railway Network

Since the beginning of the industrial revolution, the rail networks have been one of the main drivers of global economies (To 2015). Railway operations can be defined as "a routine vehicle movement between two or more points at pre-scheduled times, transporting passengers or cargo" (Banerjee et al. 2020). The foundation of a railway system is created through a thorough planning process that yields a timetable and resource responsibilities. Pre-planned timetables, often formed long time ahead, determines the specific time slot of the trains' arrival and departure as well as the route choice of the trains through the infrastructure. The sequences of train operations can be assumed to be stochastic processes (Şahin 2017). Rail transportation provides several advantages over other modes of transportation, including the ability to move huge quantities of people or cargo rapidly, efficiently, competitively, and environmentally friendly (Profillidis 2006). Demands for quality and accuracy in rail are rising across the board, from the government to the freight industry to the general public.

Many railways are currently running at high-capacity utilization levels as a result of decades of rising supply and demand. Railway networks are highly complex, interconnected systems where many train units share a small infrastructure of rails. As a result, railways are more susceptible to unforeseen circumstances and changes leading to deviation from the original timetable and unmet resource commitments. Due to the high interdependency between trains from the shared infrastructure, constrained timetable, and high-capacity utilization, railway operations experience scheduled waiting times and delay propagation, that grow across large distances (Goverde 2005a). The perceived level of service and unreliability of rail systems are often cited by many potential train passengers as reasons not to use the train (Cacchiani, Huisman et al. 2014). Therefore, more reliability and service will be required to entice these potential passengers to utilize the train. Using technology to increase system efficiency is one option to boost capacity and quality without making substantial infrastructure expenditures (Lüthi et al. 2007). For proactive and anticipatory regulation of the current railway traffic control and information provision in passenger rails, accurate real-time train delay prediction is a key requirement (K. Tiong et al. 2022).

2.1.1 Train Delay

Trains operate strictly according to the instances in the predetermined urban rail transit schedule. However, the real operating process frequently experiences the interaction of unpredictable factors. There are several errors, which could be caused by stochastic equipment failure or other human and operation factors. If these interruptions surpass the timeline tolerance, delays will occur. The delay of a train is calculated as the difference between the event's realized timestamp and its planned scheduled time (Thomas Spanninger et al. 2020a). In other words, delay is measured in units of time. Delays can be either positive or negative, indicating late or early arrivals and departures, respectively. Let t_D^s and t_D^r denote the scheduled and realised departure times, and d_D the departure delay:

$$d_D = t_D^s - t_D^r \tag{2.1}$$

As railway networks are heavily intertwined systems where numerous trains share a limited infrastructure of tracks, they are susceptible to delays from even the smallest deviations. Delays lead to disruption of the normal flow of the railway, resulting disorganization in the transportation system, extending passenger journey times, reducing customer satisfaction and the overall reliability of the railway (Rudnicki 1997). Delays can be classified as either a primary delay or secondary delay, here defined by Goverde (2005a):

Definition 2.1.1. *Primary delay:* A primary delay is the deviation from a scheduled process time caused by disruption within the process.

Definition 2.1.2. *Secondary delay:* A secondary delay is the deviation of a scheduled process time caused by conflicting train paths or waiting for delayed trains.

Primary delay is a delayed train directly caused by variations within a process, e.g. unusual high number of boarding passengers causing longer dwell time or extreme weather causing speed limits (Goverde 2007). Due to the high degree of interdependency in railway operations, primary delays often expand to other trains in the railway network, causing secondary delays (Jianxin Yuan and Ingo A Hansen 2007). In other words, secondary delays, also known as knock-on delays, indicate the delays caused by another train. The majority of delays arise from minor inconveniences, which mostly happen at train stations according to Palmqvist (2019). Over the course of a lengthy trip, these minor inconveniences can add up and cause significant delays. Dynamic delay propagation is the process by which a secondary delayed train causes additional delays to other trains which can influence the entire network. In order to create effective countermeasures and maintenance plans, it will be very valuable to have the ability to identify, relate, and explain the combinations of primary and secondary delays as well as how they cascade throughout the network.

2.1.2 The Hierarchical Railway Planning Process

Due to the complexity of the individual choice issue in railway planning where judgments taken at each stage will affect the next, a hierarchical strategy must be adopted according to Goverde (2005a). Complex planning is required for passenger railways, the signaling systems that must ensure safe train traffic, and which also take into account the unavoidable train delays brought on by the complex interactions between technical systems and human behavior. The many interacting processes on the modern railway depends on a wide range of factors from within the railway system, technical devices, and from exogenous sources. Variety leads to potential risks of disruptions to the traffic processes. Although each stage typically takes several months to complete, the hierarchical approach enables an iterative solution procedure where prior decisions must be adjusted to find a workable or optimal solution in a later stage.

The hierarchical railway planning process typically distinguishes between three levels of different planning horizons; *strategic planning* for processes planning, *tactical planning* for control of railway traffic and *operational planning* for management of railway traffic (Goverde 2005b; M. R. Bussieck et al. 1997; Cordeau et al. 1998). Train delay prediction belongs to all levels of each process. Strategic planning focuses on resource acquisition while addressing the fundamental goals of the railway undertaking. The delay prediction model supports both the planned infrastructure investment and the strategic design of the scheduled transportation network. Travel demand is converted into transport supply, train lines, and traffic means for allocation to the transportation services during the strategic planning phase (Goverde 2005a). Tactical planning phase involves creation and analysis of the timetable, concerning with capacity allocation of resources to transport services. Timetables are developed and updated annually or seasonally, specifying train routes and schedules. To create feasible timetables, accurate delay prediction is essential. Operational planning refers to day-to-day activities that address scheduling changes during an operation in the event of unforeseen circumstances. Timetable optimization, conflict detection and resolution, quality of service enhancements, and delay recovery are all critical issues that receive top priority during this phase (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019b). In order to make real-time decisions and dispatch disruption trains, operational level processes depend on accurate predictions of train movements (Goverde 2005a). Real-time decisions are essential regarding rescheduling, train re-sequencing, or rerouting, as well as to provide reliable passenger information to passengers for trip planning (K. Y. Tiong et al. 2023a).

Successful planning on all three process levels is crucial for an optimal railway system, and can be characterized by flexibility and adaptability (Profillidis 2006). The railway planning process is repeated yearly so that previous year's solutions can be assessed and used again or improved in the current designs. Therefore, having access to accurate empirical data to compare the schedule design and its reality is an important factor to consider when creating a new annual timetable as well as when identifying and managing deviation between plan and realization during daily operations. Although train delay forecasts can drive strategic, tactical, and operational level applications, most studies concentrate on creating predictive models to aid decision-makers in creating efficient management strategies, which belongs to the operational planning phase (K. Y. Tiong et al. 2023a).

2.1.3 Train Delay Prediction

Accurate real-time predictions of train delays are crucial for proactive and anticipatory regulation of the existing railway traffic control. Additionally, train delay predictions are important service qualities for passengers reducing the uncertainty about their arrival and departure times, allowing them to take proactive actions to lessen the impact of train delays (Spanninger et al. 2022).

The train delay prediction methods can be categorized into event-driven and data-driven models based on their inherent modelling paradigms (Wen, P. Huang, Zhongcan Li and Mou 2019). In the recent study by Spanninger et al. (2022), a brief literature review of both approaches is provided. Event-driven approaches explicitly capture the interactions between train-events, which entails the construction of a network of consecutive train events such as arrivals, departures, and pass-through. This approach aims to model the procedures and restrictions governing rail operation dynamics. The event-driven approach is an iterative process with multi-step predictions since predicting train incidents in the near future directly affects projecting delays that will occur later. Most event-driven approaches produce stochastic predictions. Event-driven approaches are primarily based on either a graph model or an equation system, such as Markov Chains, Petri Nets, Bayesian Networks among others (K. Y. Tiong et al. 2023a). Data-driven methods typically generate one-step predictions by mapping the input to output, at the desired station or point in time, without using the intermediate predictions based on train-event dependency structure (Spanninger et al. 2022). Data-driven methods use observed data to identify useful feature sets and decision criteria.

Most data-driven models are statistical, machine learning, or deep learning models, typically yielding deterministic predictions. According to K. Y. Tiong et al. (2023a), some models are more popular than others, with Neural Networks and Random Forest models being the most widely used. All data-driven models are totally dependent on the availability of high-quality historical data. Hence, the methods are very popular due to the significant increase in data accessibility and availability, especially in the railway system domain, which improves model calibration and test performance. According to the thorough literature surveys by Ghofrani et al. (2018) and Wen, P. Huang, Zhongcan Li, Lessan et al. (2019b), data-driven approaches have been widely adopted for railway operation, maintenance, and safety since 2010. The surveys elaborate upon how research from the literature demonstrates the benefits of utilizing big data analytics in rail transportation systems to reduce costs and delays while continuously retaining high standards for reliability, safety, and customer satisfaction.

Data-driven train delay prediction models can be categorized into long-term and short-term delay prediction models, according to Faverges et al. (2018). *Long-term delay predictions* are used at both strategic and tactical levels where historical train operation data, including weather forecast, public holidays, and seasons, among other external factors, is considered. When different scenarios are encountered, train operators can make the necessary adjustments to their plans days or even months in advance. *Short-term delay predictions* are used at the operational level, fed with real-time data of the current delays throughout the network with the goal of estimating delay at the next stops. As stated by Spanninger et al. (2022), the prediction accuracy of a prediction model decreases as its prediction horizon is extended. Furthermore, short-term prediction models are frequently developed to forecast train movements on a particular train line, indicating that the model has a limited capacity to generalize to other train lines. However, through updated, real-time information, network insight and "hidden" knowledge extracted from train operations, short-term data-driven models can support train dispatching decisions and activities (Faverges et al. 2018).

2.1.4 Explanatory Variables

Understanding the existing impacts of the factors that cause congestion and disturbance on the railways is crucial. In the recent study by K. Y. Tiong et al. (2023b), a categorization of explanatory variables of train delays in existing literature is presented; namely train operation variables, network variables and external variables. An overview of examples of the explanatory variables from current literature can be found in table 2.1.1.

Train variables, such as train and train event operations, are variables that capture the variability of railway traffic. The type of train, its structure and design constitute specific requirements for their circumstances. Different trains require various amounts of infrastructural capacity, operate at different frequencies, and have different priorities. This, for instance, can be seen on the required train dwelling time. Wider doors (Perkins et al. 2015) and a lower train floor to reduce the vertical gap (Holloway et al.

Variable	Type of feature	Concrete examples
Train Operation	Train	Train type, train design (length, tonnage, door width etc.), train speed and horsepower, train count, train direction, train priority and train order
	Train Events	Delays (arrival, departure, run-time, dwell times, headway), dwell time (scheduled and actual), run time (scheduled and actual), running time between stations, buffer time, headway (scheduled, actual), train interaction(meet, pass, overtaking)
Rail Network	Location	Infrastructure availability, topography, demand for rail service, attributes (station, area), distance travelled, percentage journey travelled
	Infrastructure	Number of tracks, occupancy of track segments, conflict indicators for track occupation, track allocation, platform conflict indicators, designated platform, platform change status, and the quantity and accessibility of sidings
External	Calendar	Time of day, days of the week, months of the year, weekends or workdays, time of year, holiday, season, peak hours or off-peak hours
	Weather	Temperature, wind direction and speed, snow depth, and precipitation or rainfall
	Disturbances	Duration of delays, reasons for disturbances, intensity of disturbances, duration of disturbances, the number of affected trains
	Others	Planned maintenance, customers, train crews

Table 2.1.1: Explanatory variables in previous research, categorized by K. Y. Tiong et al. (2023b)

2016) reduces dwelling time. On the other hand, longer trains remain in place for a longer duration because train operators need more time to ensure no passengers boarding before departure (D. Li et al. 2016). Also, the acceleration and speed limits of various train types may vary, which may impact how they operate. Train event variables refers to all the pre-scheduled and actual train activities and movements. As mentioned by K. Y. Tiong et al. (2023b), train event variables have been identified as the primary factors causing delays in railways. Train event variables are therefore crucial when studying train delays and an important variable in train management models (Corman and Meng 2015; Wen, P. Huang, Zhongcan Li, Lessan et al. 2019b).

Rail network variables includes the infrastructure and locations of the physical environment for train movements. Infrastructure variables encompass the equipment designed to support, its structure and buildings of the railway system. In contrast to other means of transportation, the railway network is an interconnected system where the infrastructure is often scarcely available and rarely offers viable alternative route between any two points (Dekker et al. 2019). Both small and large deviations can result in delays. A delayed train must use infrastructure at times that weren't anticipated, which creates a conflict when two or more trains requests the same element at the same time. Infrastructure failure, more specifically issues with security, signalling, and track systems, were the primary cause of delays on the Norwegian Railway in 2021, accounting about 20-30% of the total delay (BaneNOR 2023e). Location variables relate to variables that define a train´s position on the train network. To identify the pattern of train delay throughout its route, the train operation conditions on each location needs to be mapped.

The external variables include calendar, weather, disturbance, and others. The rail demand relies on the calendar; if there are any holidays, whether it is a weekday or a weekend, and what time of day it is. According to N. Olsson and Haugland (2004), most delays occur during passengers boarding and disembarking, where the actual dwelling time exceeds the planned. Weather variables, especially extreme weather have a negative impact on the train punctuality (Zakeri and N. Olsson 2018; Ling et al. 2018; Koetse and Rietveld 2009a). Other external variables can be variations in internal and external factors on daily operations, which results in different individual process times on the same operation (Goverde 2005a).

A train delay's root causes are frequently numerous, making the causal picture extremely complicated. However, a prediction model will not consequently be more accurate the more variables it is fed. According to Thomas Spanninger et al. (2020b), real-time data are an essential input in prediction models whenever methods are used to predict delay development in real-time. Further, most studies take incorporate observations of realized historic train events, followed by location and external variables. An overview of previous literature focusing on train delay prediction approaches and the input data can be found in both K. Y. Tiong et al. (2023b) and Thomas Spanninger et al. (2020b).

2.1.5 Punctuality in Railway

In rail services, punctuality is a key performance indicator that gives insight into whether the planned schedule is optimal or not. Rudnicki (1997) defines punctuality in transport services as “a feature consisting in that a predefined vehicle arrives, departs or passes at a predefined point at a predefined time”. It is used as a fundamental indicator of the reliability of railway operations, assessing an entity’s capacity to carry out a specified task under specific operational and environmental circumstances for a predetermined amount of time (Jianxin Yuan 2006). Several studies point out punctuality as a critical significant quality factor of the railway system, maybe the most important in railway operations (Parbo et al. 2015; Joborn and Ranjbar 2022; Økland and N. Olsson 2021). Weather, the capabilities of the rolling stock, driver behaviour, traveller behaviour, and even the quality of timetables may have an impact on train punctuality, which affects how efficiently capacity is used and how stable the operations are (Lee et al. 2016; Sameni, Landex et al. 2011). Deviations from the origin plan reduce the overall level of service. Punctuality can be used to evaluate the level of a product’s perceived performance, quality, and satisfaction, and determine whether the infrastructure can still maintain connections under adverse conditions.

In response to the rising demand, the capacity utilization of the rail network has increased. Due to the constrained track capacity in highly interconnected train networks, which are common throughout Europe and are particularly prevalent in large cities, a train’s punctuality is crucial for every other actor in the network. To maintain the performance and competitiveness of railway enterprises, improved punctuality is required (Zakeri and N. Olsson 2018; Kjøsnes 2015). To achieve high punctuality, the rail operating firms are prioritizing the prevention and mitigation of delays (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019a). The literature has long acknowledged the trade-off between improving the dependability and punctuality of train operations and expanding the utilization of the railway network (Jianxin Yuan 2006). In order to utilize track capacity to the fullest while maintaining the necessary level of punctuality, it is crucial to get a realistic picture of each process on the rail network. In order to maintain current ridership levels and draw in new ones, scheduled train services must also ensure high punctuality. Customer satisfaction, applied in the commercial sector, is correlated with the apparent gap between real and ideal service performance levels (Stradling et al. 2007).

To offer the best user experience, it is necessary to use customer satisfaction measurements. Several studies have shown that punctual trains are requested by passengers above more frequent trains (Lam and Small 2001b; Norheim and Ruud 2001; Nyström 2008). Travellers are willing to pay more for reduction of variability in their travel time, according to Zheng Li et al. (2009) research. Low punctuality can be described as a “dissatisfier”, meaning that the lack of the quality aspect according to the expectations has a negative rating (Rudnicki 1997). On the other hand, a sense of control and satisfaction over the travel experience can be attained with punctuality. The study by Lam and Small (2001a) quantifies the impact of the "dissatisfied" on both time and reliability values. They claim that rather than the actual travel time, travellers are more interested in the reliability of the transit time. These unit values are frequently used when estimating the benefits of government initiatives. The values vary depending on several factors, and studies have indicated that the unit values tend to be higher in large cities and rise with travel distance. The table below, 2.1.2, lists some suggested values for travel time on board regularly operating trains for a trip under usual circumstances.

Trip purpose and distance	Under 70 km	70 - 200 km	Over 200 km
Business	451	391	419
Commuting	108	183	233
Leisure	94	120	150
All purposes *	109	162	193

Table 2.1.2: Values of travel time in rail transport, by trip purpose and distance (NOK per hour) (Lam and Small 2001a)

The focus on improving punctuality has increased over the last decade. The European research and innovation program Shift2Rail, Europe´s Rail (2020), was established in 2014. Their vision was to deliver, via railway research and innovation, the tools necessary to create the most competitive, time-driven, cost-effective, high performing, sustainable, and environmentally friendly mode of transportation for all of Europe. European Commission (2021) worked towards increasing punctuality by 50% by 2020 as one of the overall targets.

2.1.6 Punctuality and Delay

Although punctuality and delays are sometimes used interchangeably, there are some variations between the two. While punctuality is more of a numerical indicator of operational reliability expressed by the percentage of trains arriving at stations on schedule, delay is measured in time unit (Økland and N. Olsson 2021; Parbo et al. 2015). Both measures are commonly used in the analysis of the railway network. However, the punctuality rate doesn't say anything about the each individual railway operation, but rather gives a reflective review of the overall reliability. According to a recent article by Palmqvist and Kristoffersson (2022), there is a discrepancy between the desired level of high train punctuality and the parameters that are conventionally measured to affect punctuality, such as elements of categories including infrastructure, operations, timetable, and weather. These parameters are the same ones causing delays.

Since delays are frequently inevitable, transit planners and specialists have worked hard to improve the performance and reliability of public transportation networks. Due to its importance, punctuality is a common measurement in railway services. In rail, punctuality is the proportion of trains that arrive specific position within the time threshold (Cerreto et al. 2016a). The punctuality goals vary per nation and may vary between regional and long-distance service, and the category of the journey or travel time (Blayac and Stéphan 2021). Table 2.1.3 shows the delay thresholds for rail services in Europe. The Norwegian thresholds are bold since this paper is based on data from the Norwegian railway.

The punctuality criteria are satisfied if the train arrives at its defined destination within the allowed margin. The Norwegian standards, according to the punctuality report from 2021 (BaneNOR 2023e), punctuality is the percentage of trains that arrive at the destination and Oslo S within a frame of 03:59 minutes. The margin is 05:59 minutes for freight trains, cross-border trains, and long-distance trains. Delays that are recovered and fall within the acceptable range for punctuality are not included in the punctuality statistics.

Delay Threshold	Regional Service	Long-Distance Service
More than 30 s	Hungary	Hungary
More than 1 min	Croatia	Croatia
	Japan	Japan
More than 2 min and 30 s	Finland	
More than 2 min and 59 s	Denmark	Denmark
	Switzerland	Switzerland
More than 3 min	Spain	Spain
	Netherlands	
More than 3 min and 30 s	Latvia	Latvia
More than 3 min and 59 s	**Norway**	
More than 5 min	Bulgaria	Bulgaria
	United-Kingdom	
		Netherlands
	Poland	Poland
	Portugal	Portugal
	Slovakia	Slovakia
More than 5 min and 29 s	Austria	Austria
More than 5 min and 59 s	Germany	Germany
		Norway
	France	France
	Belgium	
	Sweden	Sweden
More than 10 min		United-Kingdom
More than 10 min and 59 s		Belgium

Table 2.1.3: Delay thresholds for regional and long distance services in Europe. Source: Blayac and Stéphan (2021)

2.1.7 Railway Timetable

A railway system is a public transportation network based on periodic schedules. Timetable planning falls under the category of tactical level planning. Planning for transportation entails choosing the best routes between an origin and a destination, and allocating the required resources for the future (M. Bussieck 1998). An essential component for the efficient functioning of a railway network is the comprehensive schedule of activities. Railway scheduling aims to establish a schedule for trains running on a railway by identifying their movements on the network in space and time. All events on a properly specified timetable have planned timing, which is first established by the realistically scheduled process timings for the various processes (Goverde 2005a). It outlines the routes that the trains will take on the railway network, including the track lines, the stations and junctions that they will pass through, connections between trains for passengers or freight, and interactions between trains for safe operations (Harrod 2012a). Caprara et al. (2002) and Schlechte et al. (2007) provide comprehensive discussions of several models and solution approaches for developing railway timetables.

The problem of scheduling railway operations, also known as railway timetable planning, is known to be an NP-hard problem (Mascis and Pacciarelli 2002). However, it is one of the key factors in the successful operation of a railway network. Large-scale railway networks are challenging to schedule, especially in approaches which model the railway network at a high level of detail. Individual process times are never exactly the same from hour to hour or day to day because of the variations in internal and external factors (Goverde 2005a), even though the railway system in based on periodic schedules of operations. Modern and complex railway systems, where the infrastructure is used for both passenger and freight services, in addition to maintenance services, every day of operations is unique. The study by Gestrelius et al. (2015) identified 314 different patterns when considering intervals of entire days, studying historical data from a whole year of 365 days. Further, when considering smaller intervals, a higher amount of the same patterns repeated is expected (K. Y. Tiong et al. 2023a). However, railway operators want to resolve complex network-wide scheduling issues quickly and to a high standard in order to increase the capacity of the railway for transport (Leutwiler and Corman 2022). It is tempting to run trains at full capacity on the lines while ignoring the need for margins and supplements, which take up capacity as well. The outcome is a schedule that is susceptible to disturbances. Hence, railway schedules are designed capable of withstanding changes in operating conditions, disturbances, and delays without compromising functionality.

2.1.8 Robustness in Timetables

Robustness in timetables is one of the key factors in the successful operation of a railway network. The robustness of a system or plan describes how it responds to uncertainty (Lusby et al. 2018b). Robustness is the capacity to maintain certain performance characteristics, such as stability and performance to withstand threats and interruptions (Hong et al. 2019). It is considered as the sensitivity to disturbances with stochastic variables. Given that schedules are set plans, it can be challenging to make them robust enough to withstand various unforeseen circumstances without significantly compromising their effectiveness (Kauppi et al. 2017). One of the primary factors that can have an impact on the whole railway system is a timetable defect, whereas one train's delay can have an impact on trains downstream due to the network's interaction. The distribution of delays provides a clear picture of how robust a timetable's design is and how stable train operations are. Secondary delays often arise during their arrival or departure at stations where the crossing and/or merging of lines and platform tracks functions as bottlenecks, especially in extensively utilized railway networks (Jianxin Yuan and Ingo A Hansen 2007).

Network planning and management now place a high priority on building robustness in transportation networks (Harris et al. 2013). Robustness has been used in numerous transportation industries to examine the network performance of the transportation system during disturbances. Among other research, robustness was studied by Hong et al. (2019) and Ye and Kim (2019) in railway system, L. Zhang et al. (2013) and Tang and S. Huang (2019) in urban road network, Sun et al. (2018) and X. Yang et al. (2018), in urban rail system, and in air transport by X. Yang et al. (2018). Both Cacchiani and Toth (2012) and Lusby et al. (2018b) reviews models for including robustness in rail operations. They demonstrate how issues of robustness in the quality of particular planning problem solutions and of operations in general have received increased attention. The quantification of performance may be used as a robustness indicator for a wide range of diverse challenges in railway planning and operations. These issues arise in practice, starting with issues with strategic planning and extending to more operational level planning. Network planning, line planning, timetabling, rolling stock planning and crew scheduling are all individual operational planning problems which can benefit from high robustness.

Timetabling is the area most studied for implementing robustness (Lusby et al. 2018b). Many studies have been conducted on how to construct a schedule that is more robust to delays (Joborn and Ranjbar 2022; Andersson et al. 2015; Cerreto et al. 2016b; Solinen et al. 2017). Every underlying process' stochastic behavior has an impact on the robustness of the schedule. To prevent quick delay propagation, the unforeseen variation in process time must be taken into account (Goverde and Ingo A Hansen 2013). The distribution of buffer times is a key factor in many robustness concerns. The appropriate level of time supplements and buffer times must be determined using analytical, stochastic, and statistical methods without being too cautious.

2.1.9 Timetable Supplement

Timetable supplements, often referred to as slack or buffer times, are frequently utilized tools that add buffers and reserves, respectively, to minimum process duration and minimum headway between train tracks in scheduled timetables to lessen the susceptibility if deviations occur (N. Olsson, Halse et al. 2015). These deviations may be caused by disturbances, but they may also include variations in the time used for certain activities, such as dwelling time on a station, and differences in train speed depending on the driver. These uncertainties around the exact running and dwell times of trains are reduced by buffer times. Buffer time may be regarded as a directly relevant or quantifiable indicator of robustness for a modest departure from anticipated running times. The study by Dewilde (2014) demonstrates how establishing robustness in rail timetabling relates robustness to delay propagation. More "robust" timetables are ones that are less vulnerable to propagating delays, and buffer times are virtually always connected to absorbing delays and then preventing their further propagation.

Jianxin Yuan and Ingo Arne Hansen (2002) used a stochastic model for train delay propagation analysis at stations on the Hague Holland Spoor in the Netherlands to demonstrate the necessity of buffer time in their study. The research showed that when the predicted buffer time between trains at level crossings decreases, the average knock-on delay of all trains increases exponentially. It was noticed that a train timetable's distribution of buffer time affects the probability of interference and has a significant impact on resolving and minimizing train interference.

The optimization of the running time allocation and supplements in connection to the expected operating cost and effectiveness of train operations has garnered a lot of attention in the literature (Jianxin Yuan and Ingo A Hansen 2007).

Another aspect of the effort on punctuality advancement is reducing variance. Almost all processes suffer from variation as it reduces the effectiveness of the process or degrades the quality of the final result. The negative impacts of variation are especially noticeable in the railway industry, where a lack of flexibility and complicated railway systems are issues that are closely related (Profillidis 2006). The various components that make up travel time variability include differences in travel times from day to day, throughout the day, and even from vehicle to vehicle. Even little variations in process timings might result in significant delays (Goverde 2005a). Because of the tight integration, changes in one train's movements have an impact on others. This can cause a conflict when two or more trains try to utilize the same element at the same time (Cerreto et al. 2016a), which can cause delays to spread.

Choosing how much buffer time to utilize involves making a trade-off between increasing robustness, where the probability of generating primary delays reduces and the punctuality increases, and a loss in network use and passenger service. For instance, more buffer requires fewer trains running and longer waiting time for people changing trains, increasing the travelling time and operating costs which reduce the efficiency and attractiveness of the rail. It is crucial to maintain time supplements and buffer times at a reasonable level since they consume capacity.

2.1.10 Capacity in Timetable

The capacity of a rail network is the specific amount of train operations planned on a number of infrastructure elements in the timetable under a specific service plan (Krueger 1999). Optimizing the use of railway infrastructure is a complex and difficult, but an important task. The capacity of a railway network is extremely dependent of how the network is utilized (Abril et al. 2008). Capacity is dependent on the specific composition of trains and the sequence in which they travel on the route due to the physical and dynamic variability of train characteristics. Also, it evolves as operating environments and infrastructure change. Some fundamental factors affects the railway network capacity (Abril et al. 2008). The block and signal system, single and double tracks, track structure and speed limits, among others are infrastructure parameters affecting the capacity (Borndörfer et al. 2018). Train composition, regular timetable, traffic peaking factor, among others are traffic parameters affecting the capacity (Corman and Henken 2022). Operational conditions like track interruptions, robustness, train stop time, among other parameters are all affecting the capacity.

Abril et al. (2008) examined the Spanish rail network to identify the key variables influencing rail capacity. The results show that train capacity changes depending on the speed of the train, where it stops, the distance between railway signals, and how reliable the train schedule is. Although different nations and their rail networks have different train traffic densities and timetable structures, performance measures including infrastructure occupancy, timetable stability, feasibility, robustness, and resilience apply to all railway schedules (Goverde and Ingo A Hansen 2013). The metrics provide a structure for categorizing data in order to assess the development of timetabling design processes.

However, a variety of techniques and equipment are applied in capacity utilizing of the railway network. Seven infrastructure administrators' timetabling processes were examined in the research of Goverde and Ingo A Hansen (2013). The methods used to create schedules in the past ranged from not detecting conflicts to using analytical, stochastic, and statistical techniques to create a workable and reliable timetable. An overview of research on timetable optimization and the fundamental model structures can be found in the research by Harrod (2012b).

Previous railway capacity studies are analysed and discussed in the thorough literature study by Sameni and Moradi (2022) to further our understanding of how to achieve effective capacity use. The review discusses the importance of the use of different approaches to analyse capacity. Further, capacity analysis methods are divided into three categories, major papers and their contributions are discussed and the strengths and weaknesses of each method are emphasized.

2.1.11 Railway Traffic Control

Railway traffic control, also referred to as train rescheduling, is the process of modifying schedules in real-time to minimize the effects of unforeseen events (Corman and Meng 2015). The process of adapting the plan to the real circumstances is complicated and presents many potential for improvement. In order to reduce delays and achieve punctuality targets, train rescheduling has been an important study field in recent years. The growing availability of data in the rail industry gives information from the historical values. The nature of each factor and their involvement are crucial information in the forecasting process that must be taken into account (Petropoulos et al. 2022). Conflicts must be resolved immediately in the event of an unplanned disturbance that prevents the timetable from being met. Disturbances are comparatively minor perturbations to the railway system that can be managed by modifying the time schedule without changing the responsibilities for the crew and rolling stock (Cacchiani, Huisman et al. 2014). Disturbances corresponds to a rail process, e.g. dwelling in a station or driving between two, that takes longer time than specified in the timetable. Railway traffic control focus on minimizing delays of trains and considering precise movements of train vehicles which minimize travel time of passengers, and consider larger scale networks by which less train traffic is modelled very roughly. Railway traffic control attempts to provide an operation-centric strategy, minimizing train delays and taking precise movements of train vehicles into consideration (Corman, Pacciarelli et al. 2015). Additionally, it provides a passenger-centric strategy that reduces travel time for passengers and takes into account bigger scale networks with less accurate train traffic modeling. In both operation- and passenger-centric systems, delay prediction is essential. The ability to predict delays accurately is crucial for dispatchers to make smart decisions and for the operation of trains (Wen, P. Huang, Zhongcan Li and Mou 2019). The fundamental idea behind forecasting is that predictions about the future may be made using information from the past and present (Petropoulos et al. 2022). There is a conviction that previous value patterns may be found and successfully used to forecast future values, especially for time series. The method is thought to be extremely difficult, requiring different constraints for every situation when a problem arises (informs 2022), yet it is necessary to lower the level of uncertainty in railway dynamics. If you can adapt the plan to the current circumstances, accurate forecasting offers great opportunities for improvements. The accurate forecasting of future values, however, is anticipated to be a challenging undertaking.

Train dispatching is highly regulated, as is any management of essential infrastructure. There are highly trained human dispatchers that are responsible for the decision making in the current railway system (Mannino and Nakkerud 2023). The problems are considered as multi-criteria decision-making problems (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019b), where the whole network must be taken in to consideration. The timing of train arrivals and departures, their route, and the arrangement of trains on common track sections are frequent considerations that must be taken into consideration. In reality, these choices are made using a combination of common sense, dispatcher expertise, and real-time data. The dispatchers must make decisions rapidly while taking into account the system's present status as well as prospective temporal and spatial state changes. In order to increase the overall transport efficiency of the railway system, operators must be able to accurate delay predictions in order to quickly identify recurring delays on railway routes and potential conflicts in train operation. As a result, rather than focusing on offering the most ideal solutions, issue formulations and solutions may instead be heuristic, sub-optimal ones (Filcek et al. 2021). Using apps that forecast train delays, where decisions are represented using a variety of factors and restrictions, can ease the complexity of train rescheduling in real-time railway systems. Researchers have investigated the use of models and algorithms to improve the quality of the railway services offered, which has raised the usage of the involved railway systems (Cacchiani, Huisman et al. 2014). An integrated system that combines train control and real-time railway traffic control was recently tested on a section of Swiss railway as part of the study by Lüthi et al. (2007). The study demonstrated how early detection of a delay may considerably minimize the overall system latency. It demonstrated how the specific timetable and network infrastructure had a significant impact on the approach's efficacy. The study also shown that quality may be increased if trains can be timed extremely accurately, for instance by departing a station at a very exact point.

2.1.12 Train Dispatching Issues

The development of decision support tools by utilizing the most recent computer techniques has drawn the interest of academic researchers as well as operational operators or modelers in the rail transportation sector. To assist judgments and actions in the face of delays in future train operations, hidden knowledge gathered from railway operation data can be used. The existing methods and models used for train delay analysis may be divided into three categories: delay distribution, delay propagation and train rescheduling (J. Wu et al. 2022). Delay distribution helps dispatchers get the probabilities and length of delays by providing a train with basic principles. The task is difficult due to the varying operating conditions of some lines and networks, which have a substantial influence on train delay distributions. Dispatchers may use a number of strategies to absorb delays when a train is delayed, including reducing running speeds, changing dwell periods, and changing overtaking (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019a). Delay propagation is a function of delay aggravation caused by disturbances and delay recovery activities. Traffic controllers are unable to forecast the spread of delays, particularly when there are complicated rail networks, dense traffic, and significant disturbances (D'Ariano and Pranzo 2009). The reduction of delay propagation, however, is given top importance. One of the most significant and challenging research questions in delay propagation is believed to be delay prediction (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019a). The timetable needs to be rescheduled by the dispatchers when delays cause disruptions in train operations. Train scheduling is frequently used to describe the decision to change the arrival, departure, and operating hours of subsequent trains. Via a number of topics, including delay recovery, conflict detection and resolution, schedule optimization, and quality of service enhancements, the main challenges of timetable rescheduling are reviewed. The "Handbook of Optimization in the Railway Industry" by Borndörfer et al. (2018) provides more descriptions of the train dispatching problem and a summary of the many model types and solution strategies established over the years.

2.2 The Norwegian Railway Network

The Norwegian rail network consist of 28 rail lines which together constitutes a total of 4221 km in line kilometers with 335 train stations and stops (Jernbanedirektoratet 2023). The railway network is mostly single track with crossings at stations and certain critical junctions. Double tracks have been established on most of the sections closest to the capital Oslo, which constitutes 290 line kilometers of the rail network. There are parallel railway lines on only a few sections. The network is arranged in a star pattern around Oslo, with lines running through it from north to south and east to west. This area distribute the highest passenger volume and the railway capacity utilization is maximized with the most departures (Sørensen, Andreas Dypvik Landmark et al. 2017). Over 70% of the rail traffic goes through the Greater Oslo, making it a high-priority area. Any issues here might have an impact on all of Eastern Norway and the rest of the railway system. The utilization decrease with distance to the capital. Most regional, long-distance, and freight trains run on single-track lines. Commuter trains running through Oslo runs often on both single- and double tracks. Figure 2.2.1 gives a roughly illustration of the railway network in Norway. A comprehensive railway map can also be found in appendix A.

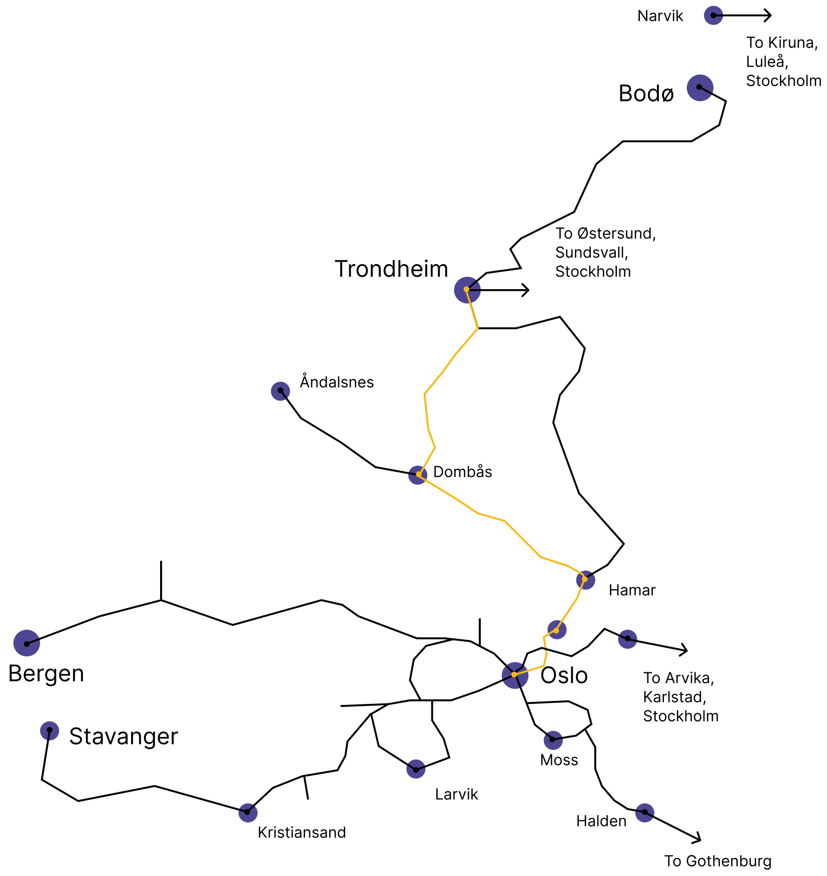


Figure 2.2.1: The Norwegian Railway Network. The yellow line symbolise the chosen train route, Oslo-Hamar-Dombas-Trondheim, analysed in this thesis.

The many responsibilities of the Norwegian railway industry have been distributed among a number of actors. Today's railway system is a result of a partnership involving many stakeholders. The Norwegian Railway Authority oversees the transportation system to guarantee that it is secure for passengers and the surrounding regions. Several private and public train operators operate the trains. The state-owned company Bane NOR is responsible for the national railway infrastructure; the operation, renewal and maintenance of the train traffic in Norway.

In 2019, before the pandemic, Norway records around 81 million passenger rail trips (Sentralbyrå 2023). Growth in passenger train journeys depends on satisfied travellers and a growing customer base for the train companies. High punctuality and regularity create trust with the customers and contribute to more people taking more trains. The availability of sustainable spending options, universal station design, and rapid, accurate customer information across all channels are just a few strategies that assist to satisfy rail passengers.

Punctuality stands out as one of the most important key performance indicators across the actors in the rail industry. In Norway, passenger trains that arrive at the terminus and Oslo S within a margin of 03:59 are considered to be on schedule. For long-distance trains, freight trains and trains that cross the border between Norway and Sweden, the margin is 05:59 minutes. Bane NOR records the punctuality data for each running train. The national database TIOS records remote-control systems' line data and dynamic train movements. The time registrations are often based on automated registrations and are thought to be precise measures of the times that trains enter and leave various parts of the network, with a high resolution (measured and recorded to the second). TIOS provides operational and administrative assistance for users, train information, rail operations, and Bane NOR management. According to the Norwegian notification duty, all businesses using real-time systems are required to make the real-time data accessible at the national level (Vegdirektoratet 2019). Data on punctuality includes train information, station information, and codes that explain any delays that exceed the defined margins for the different classes of traffic.

2.3 Weather and the Railway Sector

Due to the unpredictability of weather conditions, particularly the extreme weather events brought on by climate change, transportation networks are currently experiencing unprecedented challenges. Weather not only directly harms transportation infrastructures, but also disrupts transportation services, which has indirect economic consequences. The railway system are thought to be the least weather-sensitive of the transport modes, often expected to run when other forms of transport are disrupted (Xia et al. 2013). However, weather conditions can influence the railway traffic and cause delays. Especially, weather impacts everyday operations on congested networks and can lead to substantial delays. Snow and ice can cause tracks to become slippery and reduce the ability of trains to accelerate and brake. This can lead to delays, slower speeds, and longer stopping distances. High temperatures can cause tracks to expand, which can lead to buckling or warping. Trains may also need to slow down due to the risk of derailment. Heavy rain can cause flooding and washouts, which can lead to track closures and delays. Strong winds can cause trees and other debris to fall onto tracks, which can cause disruptions to train services. Additionally, extreme weather can affect affect the performance of the train operator directly or indirectly through disturbances in the railway infrastructure.

To mitigate the weather-related impacts, many transport companies monitor weather conditions closely and take appropriate action when necessary, such as reducing speeds, suspending services or implementing alternative routes. Additionally, railway companies can install heating, cooling, and drainage systems to help cope with extreme temperature fluctuations, precipitation, and flooding. The effect of weather conditions on daily travel activities may change over time. One direct result of the unpredictable occurrence of severe weather is that transportation system punctuality is negatively impacted, making it difficult to forecast whether scheduled service will be provided on time. Research on how weather impact the transportation sector is therefor important.

It is hard to identify the best implementation options to lessen the negative effects of disruptions, despite the fact that it is widely acknowledged that transportation planning and management organizations must increase system resilience in order to minimize losses and disruptions (Koetse and Rietveld 2009b). This is partly caused by an inadequate knowledge of how various weather phenomena and specific extreme weather occurrences affect the functionality of transportation systems. Uncertainty exists over how affects vary across various event types, as well as regarding how they vary both geographically and temporally for various means of transportation within the same event. Therefore, it's crucial to provide a comprehensive picture of their effects, map all essential steps toward increased network robustness and resilience, and find chances for mode substitution (Stamos et al. 2015).

As stated by Sabir (2011), the terms weather and climate are frequently treated as synonyms even though they are quite different. Weather is the short term state of the atmosphere at a specific time and place. On the other hand, climate is the long-term manifestations of weather and other atmospheric conditions in a given area during a period long enough to ensure that representative values are obtained. Weather refers to the daily variation in the atmosphere, including the temperature, relative humidity, cloud cover, precipitation, wind, etc, while climate refers to the general conditions over long time period in a given location. More studies related to the climate changes and its impact on the transport sector have been published in the 21-century, hence the railway system (Baker et al. 2010). Further, it is becoming more and more important to comprehend how weather events affect the current public transportation networks due to changing weather patterns. There are two major divisions in the literature on weather and transportation. First, research on the effects of transportation on weather and climate changes with an emphasis on emissions from the transportation sector. See both Chapman (2007) and Koetse and Rietveld (2009a) for a briefly review. Second, research that concentrate on how the weather affects the transportation sector. This research certainly belongs within the latter category.

There are several ways to examine the impact of the weather conditions on the transport mode, according to Koetse and Rietveld (2009a). Transport systems between regions with very different weather conditions can be compared by the differences in performance. This gives an indication of the potential weather impacts. However, there are several factors that influence the differences between different regions, of which weather one factor among many others. Given the variations in transportation infrastructure, traffic operations and the operational conditions, the impact of weather on travel behavior also varies by nation. Seasonal weather variations influence on travel behavior and transport performance can also be examined. According to Sabir (2011), weather has an influence on individual travel demand. Travellers have availability to information on the exact weather conditions and advanced weather forecasts, which seems to have an impact on their choice of transportation mode. The study indicates how adverse weather has a significant modal shift in favor of bicycle, vehicle, and public transportation. In extremely warm weather, bicycles are used instead of cars and public transportation. Cycling is replaced by more public transportation and walking during really cold weather. The weather condition also seems to have an impact on the travellers behavior at the stations. Passengers seems to seek shelter from the weather and group around the same locations. As a result, it will take longer for everyone to board the train, increasing the overall dwell time.

Weather conditions have an impact on the performance of the railway system in terms of punctuality. Both running and dwelling operations are vulnerable for different weather conditions, especially extreme weather. Each railway system has its particular challenges related to the local weather. While the cold weather is unfamiliar in the southern nations, it poses a difficulty in the Northern countries. The weather-related issues can be solved in a variety of ways. The British railway system has their own "Leaf Fall Timetable Changes" (Enquiries 2023). Thousands of tonnes of leaves fall across the railway network during the autumn. The leaves on the track are compressed by passing trains producing a thin, black coating that affects train braking and acceleration. This implies that train drivers must slow down early for stations and signals to prevent overshooting them for everyone's safety. To prevent wheel spin, they must also accelerate more gradually. This results in a longer travel journey. Timetables are modified throughout the Autumn to prevent unforeseen delays to journeys.

2.3.1 Norwegian Rail Network and Weather Conditions

Norway stretches from 57° to 78° north, with mountains areas, a long coastal line and therefor a varying and regional climate. While inland regions often have harsh winters with plenty of snow, coastal areas typically experience relatively mild and wet winters. Norway has four seasons, and is strongly exposed to extreme weather. The many seasons each have their own unique obstacles for rail operations, such as early spring frost heaving, summer sun kink, and fall defoliation. These physical changes have a direct impact on the rail system. The transportation system is also indirectly vulnerable as a result of its own complex technology and strong connections to other crucial infrastructure, such as electrical and information technology networks (Langeland et al. 2021). Transport and communication systems may be disrupted by a power outage, and the traffic management system, real-time traffic, and other ICT systems may be affected by an ICT system failure. A strong and resistant rail system is necessary given the difficult weather conditions.

The Norwegian Railways has a series of procedures with the directive to stop critical conditions resulting from poor weather scenarios from having an impact on infrastructure safety. There are strategies to cope with unfavorable weather conditions that must be used in incident preparation as well as handling occurrences that result from adverse weather conditions. The instructions outline which procedures need to be followed at a certain level of the predicted and/or measured temperature, precipitation, and contribution from snowmelt. Stricter weather readiness is implemented in the case of an elevated danger of weather that is thought to be able to impair rail safety. Norway's preparation system is organized into three readiness categories: yellow, orange, and red alertness, with increasing degrees of activity. Figure 2.3.1 illustrates the emergency response system structure. The action instructions are gathered from Bane NOR´s preparedness portal, which includes tactical and operational preparedness information in emergency situations.

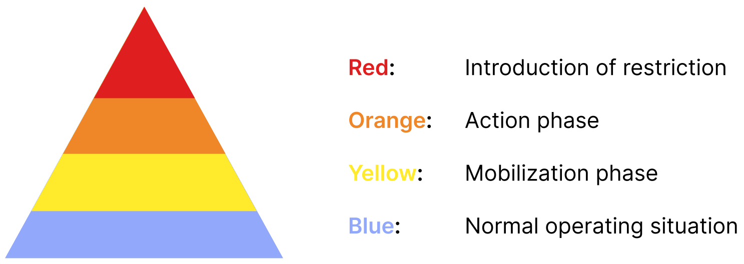

Red: Introduction of restriction
Orange: Action phase
Yellow: Mobilization phase
Blue: Normal operating situation

Figure 2.3.1: Sketch of the Norwegian emergency response system with instructions for measures in adverse weather conditions (Bane NOR 2023a)

Normal operating conditions on track sections take place when the weather situation is assessed so that safety-critical incidents will not occur. There will be a higher likelihood of potential safety-critical occurrences including solar wind, erosion damage, floods, landslides, and collapse in water-saturated embankments if the temperature and/or volume of precipitation and/or snowmelt rise over a period of 12 hours. When the weather condition is evaluated, yellow preparedness is initiated because it raises the likelihood of safety-critical occurrences. To mobilize resources for potential requirements, efforts are being made. The track section's operational environment resembles that of a normal operation level. When the likelihood of safety-critical occurrences is determined to be higher than the yellow level of readiness due to the weather, orange level of preparedness should be applied. Visibility speed must be implemented on the impacted track segment. Red level of readiness is adopted if the weather conditions worsen considerably more. At this level, the affected track section is closed.

2.4 Artificial Intelligence

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are designed to think and act like humans, including perceiving, reasoning, learning, planning, predicting, and so on (Y. Xu et al. 2021). The development of AI technology has allowed computers to perform tasks that previously required human intelligence, such as natural language processing, image recognition and problem-solving (Russell and Norvig 2010a). With the increasing power of computing and the availability of vast amounts of data, AI has the potential to transform a wide range of industries, from healthcare and finance to transportation and manufacturing. However, the development of AI also raises ethical and societal questions about the impact of AI on jobs, privacy and decision-making in sensitive areas such as criminal justice and national security.

2.4.1 The History of Artificial Intelligence

People have aimed to understand how human intelligence works for thousands of years. According to Russell and Norvig (2010a), AI is one of the newest fields in science and engineering. The science aspect involves understanding and explaining how human intelligence works, while the engineering aspect includes efforts to engineer actual intelligent entities. There is also a question of how to define intelligence. In 1950, Alan Turing posed the famous question "Can machines think?". He believed that in order to answer this question we need to define what thinking is. Due to the arbitrary nature of thought, he created a test known as the Turing test to determine if a machine can think. This test measures how closely a machine can imitate human intellect. If the machine passes the test, it can be considered as having artificial intelligence (Turing 2009).

The area of artificial intelligence (AI) is thought to have begun with a conference held in 1956 at Dartmouth College. John McCarthy, an influential figure in AI, along with other researchers from Dartmouth, brought together U.S. researchers interested in automata theory, neural nets, and the study of intelligence. During the conference, the attendees discussed the possibility of creating machines that could think and perform tasks that normally required human intelligence. Among the attendees were researchers within the field of control theory, decision theory, computer science and mathematics. McCarthy later moved from Dartmouth to Massachusetts Institute of Technology (MIT) where he made crucial contributions to the research field. He published his study in 1958 under the title *Programs with Common Sense.* The study describes his program *the Advice Taker*, regarded as the first on logical AI, i.e. AI in which logic is the method of expressing information in computer memory. The program was designed to accept new axioms and embody new knowledge of the world without being reprogrammed. According to Russell and Norvig (2010a), this represents the central principles of knowledge representation and reasoning: "that it is useful to have a formal, explicit representation of the world and its workings and to be able to manipulate that representation with deductive processes".

Over the next few decades, AI research and development made significant progress. Early AI programs, such as the Logic Theorist and General Problem Solver (GPS), were developed in the late 1950s and early 1960s. These demonstrated the potential for computers to perform tasks that were previously thought to be the exclusive domain of human intelligence. However, the scientists that developed the GPS, Allen Newell and Herbert Simon, were not content having their programs merely solve problems correctly. What interested them the most were comparing the trace of its reasoning steps to traces of human subjects solving the same problems. This introduced the interdisciplinary field of cognitive science, which seeks to construct testable theories of the human mind bringing together computer models from AI and experimental techniques from psychology into play (Russell and Norvig 2010a). Cognitive intelligence is defined as a high-level ability of induction, reasoning and acquisition of knowledge (Y. Xu et al. 2021).

Due to the successes such as the Logic Theorist and GPS fueled the optimism of AI researchers. Herbert Simon famously stated in 1957 that machines would soon be capable of solving problems equivalent to those solved by the human mind (Russell and Norvig 2010a). However, there were still significant challenges to overcome, as early rule-based programs struggled with handling complex, non-trivial problems. The key issues were the intractability of many of the challenges that AI was trying to address, computing constraints, and fundamental restrictions on the fundamental structures that were being utilized to produce intelligent behavior. The rule-based systems and symbolic reasoning were achieved by encoding human knowledge and rules into programs. In the late 1970s and 1980s, AI research shifted towards machine learning (ML), a branch of AI which involves developing algorithms that could learn from data and improve their performance over time. This change was primarily brought about by increased funding and the addition of Deep Learning techniques to the algorithmic toolset, which John Hopfield and David Rumelhart presented (Hopfield (1982), Rumelhart et al. (1986)). Another important innovation that occurred at this period was export systems. Using rule-based systems and symbolic thinking, the computers were able to make decisions and solve problems in a variety of domains. They used the McCarthy's Advice Taker approach's central idea, separating knowledge from reasoning through rules. The first commercially effective expert systems were introduced in the middle of the 1980s, and generally, the AI sector grew rapidly, going from a few million dollars in 1980 to billions of dollars in 1988 (Russell and Norvig 2010a). Following this, the 1990s revived the study of neural networks, which was initially done in 1957 in an article by psychologist Frank Rosenblatt titled *the Perceptron* (Rosenblatt 1958).

The focus of AI research changed from symbolic reasoning to probabilistic reasoning in the early 2000s. Encoding human knowledge and rules takes time, and does not allow for learning from data. Furthermore, it has a constrained capacity for dealing with ambiguity and insufficient data. On the other hand, probabilistic reasoning entails creating algorithms that can make decisions despite uncertainty and inadequate data, resulting in more flexible and precise decision-making. It currently rules AI research and enables experience-based learning.

The availability of large amounts of data has been crucial for the development of AI. The more data an AI system can be trained on, the better it becomes at the task it is trained to perform. This is because machine learning algorithms require large amounts of data to identify patterns and relationships that can be used to make predictions or decisions. With the advent of the internet and the growth of digital technologies, there has been an explosion in the amount of data that is being generated every day. This has made it possible for AI systems to learn and improve increasingly quick. The emphasis has historically been on the *algorithm*, as noted by Russell and Norvig (2010a), but new work suggests that it makes more sense to be concerned with the *data*. This is consistent with the growing accessibility of extremely enormous data collections. Banko and Brill (2001) demonstrated that bigger training data considerably improves learner performance and that learning methods should be less important in this context.

Another factor enabling AI researchers to develop more complex algorithms that can handle larger amount of data and perform more sophisticated tasks is the increase in computational power. AI models can now be trained faster and more efficiently, and they are being applied to a wider range of applications, including spam detection, machine translation and robotics (Russell and Norvig 2010a). The availability of data and computational power has also led to the democratization of AI. Instead of AI research being limited to researchers with access to specialized hardware and software, cloud-based services and open-source software has enabled a larger community of developers and researchers to experiment with and build AI models. This has opened up new opportunities for innovation and collaboration.

Overall, the evolution of AI has been marked by a shift from rule-based systems to data-driven approaches, from symbolic reasoning to probabilistic modelling, and from narrow domain-specific applications to general-purpose intelligent systems. These developments have been driven by advances in computing power, data availability, and algorithmic innovation, as well as by changing societal needs and expectations.

2.4.2 Ethical Issues and Considerations

The rapid growth of AI technology during the past decades has had a significant impact on our society. Although there are many accounts of the technological promise of AI, painting a picture of profound societal and individual change, we also find warnings about its perils.

Bostrom and Yudkowsky (2018) provides a comprehensive overview of these challenges raised by the development and deployment of AI systems. They argue that AI poses unique ethical challenges because of its potential to create systems that are more intelligent and powerful than human beings, and by such posing existential risks to humanity. A range of other issues are also raised, such as privacy, transparency, accountability and fairness. According to Bostrom and Yudkowsky (2018) the ethical concerns can be divided into three categories: (1) concerns about the behavior of AI systems, (2) concerns about the impact of AI on society, and (3) concerns about the relationship between humans and AI. Some of the issues the authors discuss within each of these categories includes the need for AI systems to respect human autonomy and dignity, the challenge of ensuring transparency and accountability in AI decision-making, and the potential for AI to exacerbate existing social inequalities.

Similarly, Russell and Norvig (2010a) list six issues concerning the development of AI technology:

- People might lose their jobs to automation
- People might have too much (or too little) leisure time
- People might lose their sense of being unique
- AI systems might be used toward undesirable ends
- The use of AI systems might result in a loss of accountability
- The success of AI might mean the end of the human race (Russell and Norvig 2010b)

It is worth mentioning that some of these threats differ little from other, less "intelligent" technologies. E.g., the Spinning Jenny and the Steam Engine reduced the need for human laborers in the 18th and 19th century respectively causing the displacement of workers. In the context of decision-making and accountability, ethical considerations should be integrated into the design and development of AI systems. AI researchers could potentially work collaboratively with experts in ethics, law and social sciences to ensure that the development benefits the society (Bostrom and Yudkowsky 2018).

One of the threats posed is particularly worthy of further consideration: "ultraintelligent machines might lead to a future that is very different from today" (Russell and Norvig 2010b). The possible consequences of AI research must be weighted very carefully so that we do not end up with a future we may not like.

2.4.3 Machine Learning

Machine learning (ML) is a rapidly growing field of study that has found widespread applications in various domains such as image recognition, natural language processing, and predictive analytics. The field is a branch of AI, and is built upon the foundation of statistical learning theory, which aims to develop algorithms that can learn from data to make predictions or decisions without being explicitly programmed to do so. The learning process in machine learning can be broadly divided into two types: supervised learning and unsupervised learning. In supervised learning, the algorithm is trained on labeled data, which consists of input-output pairs. The algorithm learns to predict the output for a new input based on the patterns it has learned from the training data.

In unsupervised learning, the algorithm is trained on unlabeled data, and it learns to discover the underlying structure or patterns in the data (Jordan and Mitchell 2015).

Progress in ML "has been driven both by the development of new learning algorithms and theory and by the ongoing explosion in the availability of online data and low-cost computation" (Jordan and Mitchell 2015). ML technology is found in many aspects of modern society, ranging from web searches to recommendations on e-commerce websites.

Increasingly, applications which employ ML systems, make use of a class of techniques called deep learning (LeCun et al. 2015). Deep learning is a sub-field of machine learning that has gained significant attention in recent years due to its ability to learn complex representations from high-dimensional data. Deep learning is based on artificial neural networks, which are composed of multiple layers of interconnected nodes, each of which performs a nonlinear transformation of the input. The process of learning the weights of the nodes in a neural network is known as back-propagation, which uses the gradient of the loss function with respect to the model parameters to update the weights.

2.4.4 Neural Networks

A subset of deep learning algorithms called neural networks is created to find patterns in data. The structure of a neural network is organized into layers, with each layer consisting of a set of nodes, or *neurons*, that perform a specific function. The neurons are designed to mimic the function of biological neurons in the human brain: neurons in one layer receive input from other layers' neurons and perform a nonlinear function on that input, which is then passed on to the next layer.

The input layer, which is the top layer of the network, receives the data and transmits it to the next layer. The network's ultimate prediction or judgment is generated by the output layer, which is the last layer. There may be one or more hidden layers between the input and output layers. The number of hidden layers can vary depending on the complexity of the task it is designed to perform. More hidden layers lead to more complex computations which in turn lead to more intricate patterns recognized by the model. The nonlinear computations from the hidden layers potentially form extremely intricate functions in mapping inputs to outputs (Jordan and Mitchell (2015), LeCun et al. (2015)). Hornik et al. (1990) showed that the multilayer neural networks are universal approximators, i.e., they can approximate any continuous non-linear function to arbitrary accuracy.

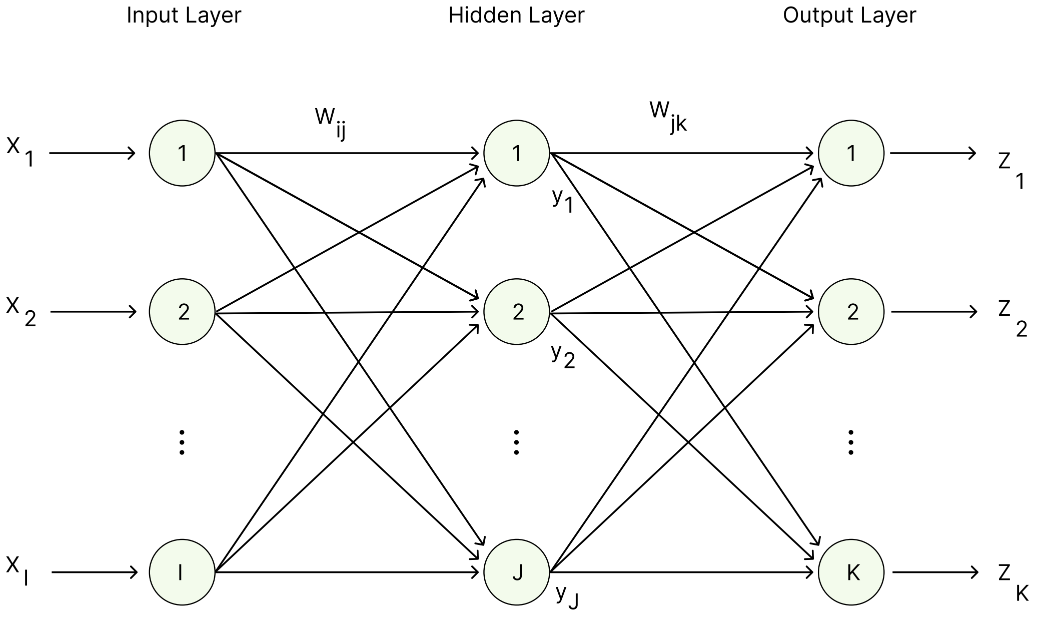


Figure 2.4.1: Flowchart of a 3-layer neural network structure

Figure 2.4.1 illustrates the flow of a simple neural network with an input and output layer, and one hidden layer with a set of neurons. The connections between the layers are associated with a set of weights, typically denoted w_{ij} from node i to node j. Biases and adjusting these weights is how the network recognises patterns in the data and eventually provide correct output through experience. These adjustments are done using a training loop, which involve two phases: feed-forwarding and back-propagation. Feed-forwarding is the process of passing an input through the network. The input data is presented to the input layer, which passes it through to the first hidden layer. Each neuron in the hidden layer performs a nonlinear operation on the input multiplied with the weights associated with each connection, and passes the result on to the next layer. This process is repeated until the output layer is reached, which produces the final output of the network. The back-propagation starts with measuring the error between the model output and the label, which is the correct value we want the model to approximate. This error value, which is often called the loss of the model, is back-propagated through the network. During the back-propagation, the weights are updated using an optimization algorithm, moving the gradient of the loss function towards minimizing the error of the model, gradually improving its predictive accuracy. The dual process of feed-forwarding and back-propagation is repeated for a number of iterations, or epochs.

Neural networks can be distinguished by the way they are connected as a network. A *feed-forward network* has connections only in one direction, which means that every neuron receives input from upstream neurons and delivers output to downstream neurons. A *recurrent network*, in contrast, feeds its outputs back into its own inputs and thus introduces loops in its network (Russell and Norvig 2010a).

2.4.5 Recurrent Neural Network

The concept of recurrent neural networks (RNN) were first introduced by John Hopfield in 1982 (Hopfield 1982), and were later brought up in 1986 by Rumelhart et al. (1986). The recurrent networks are designed to handle information from step to step by adding loops to the network topology. The loops allows the model to process sequential input more effectively rather than processing data solely in one direction. The model has the ability to take the previous outputs into account and use them to influence the current output. By conserving the node states in a hidden layer between time steps and utilizing the inherent memory produced through these feedback loops, it becomes possible to use previous inputs as knowledge to affect results in the future (Goodfellow et al. 2016).

Due to the variable duration of the input, RNNs contain a dynamic number of inputs and outputs, which makes training them different from training regular neural nets. Back-propagation Through Time (BPTT), introduced by Lillicrap and Santoro (2019), is a training algorithm used to update weights in RNNs. The algorithm works by unrolling the recurrent network and treating each time step as a feed-forward neural network with a fixed number of inputs and outputs. The network loop and its unrolling mechanism when training the network is illustrated in figure 2.4.2.

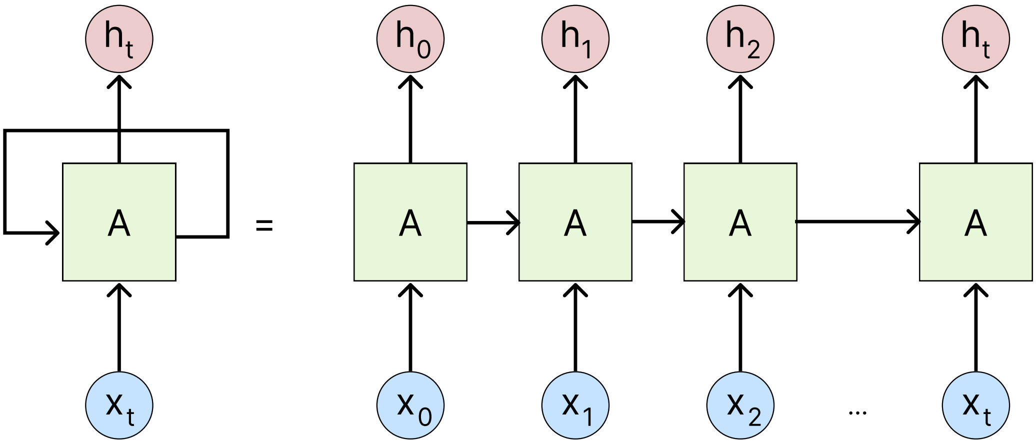


Figure 2.4.2: Unrolling the recurrent neural network

Future time steps employ the hidden layer's information in a manner similar to how human speak. Humans pick their next word in a discussion by building on their knowledge of the words that have come before it and the context of the situation. This avoids having to start considering whole new thoughts at every stage. The recurrent neural networks address this time-dependency to some extent through the ability to capture context in sequence data. Recurrent neural networks, however, struggle when there is a large gap between the present time step and the relevant information time step. The information from each time step in a recurrent neural network is stored in a vector in the hidden state. After a certain amount of time steps, some information about the input sequence is typically "forgotten". According to Y. Bengio et al. (1993) is it essential to have a deeper grasp of the issue behind the loss of information in recurrent networks in order to build algorithms that can learn long-term dependencies. Consequently this has been subject to vast amounts of research which in 1997 culminated in the paper Hochreiter and Schmidhuber (1997), presenting the Long Short-Term Memory (LSTM) network.

2.4.6 Long Short-Term Memory

The Long Short-Term Memory (LSTM) network is an advanced variant of a recurrent neural network capable of learning long-term dependencies. LSTMs has a more complex model structure, including a memory cell, an input and output gate, and a forget gate (Goodfellow et al. 2016). These gates regulate the flow of information into and out of the memory cell, allowing the model to selectively remember or forget information over long sequences of input data (Hochreiter and Schmidhuber 1997).

When a traditional RNN is trained, the network process a sequence of inputs by repeatedly applying the same set of weights to each input in the sequence. The gradients for updating these weights are computed by performing BPTT, and are multiplied with the weights at each time step, leading to the gradients becoming exponentially smaller as they propagate through time. As a result, the earlier inputs in the sequence have very little impact on the final output, and thus the network often fails in capturing long-term dependencies in the data. This phenomenon is called the vanishing gradient problem. Through the more complex model structure of the LSTM, this problem is mitigated to some extent ensuring control over the information flow of the network. The model structure is illustrated in figure 2.4.3.

As mentioned, the flow of information through a LSTM model involves four main components:

- Input gate: The input gate regulates the flow of information from the current input and the previous hidden state into the memory cell. It uses a sigmoid activation function to decide which information to keep and which information to discard.

- Forget gate: The forget gate regulates the flow of information from the previous memory cell to the current memory cell. It also uses a sigmoid activation function to decide which information to keep and which information to discard.

- Memory cell: The memory cell stores information over time and passes it on to the next time step. It is updated based on the input and forget gates, as well as the candidate values.

- Output gate: The output gate regulates the flow of information from the current memory cell to the current hidden state. It uses a sigmoid activation function to decide which information to keep and which information to discard.

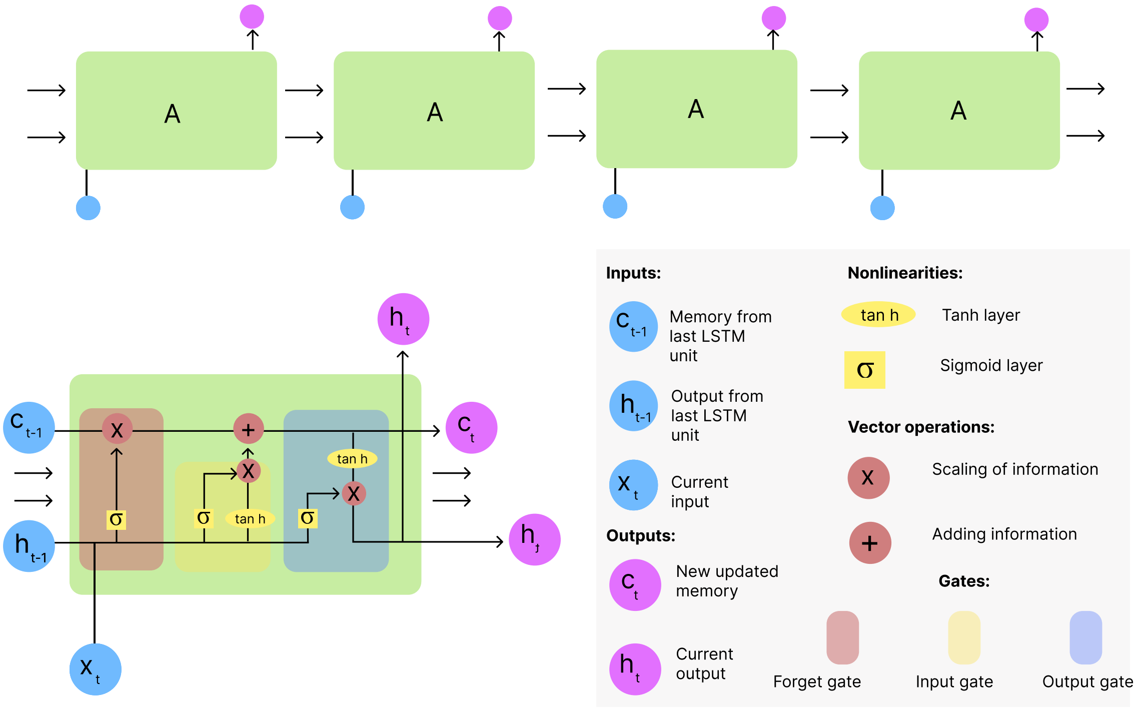


Figure 2.4.3: Long Short-Term Memory model structure

Another advanced variant of a recurrent network is the Gated Recurrent Unit (GRU), which was first presented by Chung et al. (2014). GRUs reduce the gate components from three (LSTM) to two; the update gate and the reset gate, and experiments have shown that GRU performance are comparable to LSTM (Dey and Salem 2017). Similar to LSTM, GRU uses its gates to control the flow of information and allows for the model to selectively focus on certain pieces of the information from the input data. While the reset gate determine how much information to forget from the previous state, the update gate determine how much information the current input incorporates into the new state (Goodfellow et al. 2016). This enables the model to adaptively adjust its internal state based on the current input and previous state.

2.4.7 Other Popular Machine Learning Models

As previously mentioned, according to K. Y. Tiong et al. (2023a) the most widely used machine learning models for train delay predictions are neural networks and random forest models. Random forest is an ensemble learning method commonly used for regression tasks. Neural network techniques have their counterparts in more conventional methods, including well-known linear techniques such as linear regression, and more advanced regression methods such as decision tree regression, random forest regression and eXtreme gradient boosting (Bishop 1994). This section will briefly introduce these methods.

2.4.7.1 Linear Regression

Linear regression is a statistical method used to model the linear relationship between a dependent variable y and one or more independent variables x_i.

$$y = b_0 + b_1 x_1 + b_2 x_2 + ... + b_n x_n \tag{2.2}$$

where b_i are the coefficients, or weights, which are learned during the training process of the model. Linear regression is powerful in modelling the linear relationship between variables, mostly due to the simplicity of the model.

2.4.7.2 Decision Tree Regression

Decision tree regression (DTR) is a regression method which construct a decision tree model starting with selecting the feature from the feature space that splits the data into groups with the most similar target values. Based on the values from the chosen feature the data is yet again split, and the procedure is repeated until a fully grown tree is constructed. At this point, the feature with the lowest mean squared error (MSE) for each node is selected for the split. Traversing through the tree and choosing the predicted value at the leaf node represents the final prediction of the model (Pham et al. 2017). Similar to linear regression, DTR is a simple and interpretable model, but it can however be prone to overfitting (Russell and Norvig 2010a).

2.4.7.3 Random Forest Regression

Random Forest Regression (RF) is an emsemble learning method which involve training a large number of decision trees. Each decision tree is trained on a different subset of the data. The process is repeated multiple times, and the final prediction is made based on averaging or voting the predictions of all decision trees in the forest. It is a robust and powerful method which is able to handle high-dimensional data and non-linear relationships between variables. It is also less prone to overfitting than DTR (Rodriguez-Galiano et al. 2015).

2.4.7.4 eXtreme Gradient Boosting

eXtreme gradient boosting (XGBoost, XGB) is a popular and powerful machine learning algorithm which was introduced by T. Chen and Guestrin (2016), who developed it as an optimized version of the simpler Gradient Boosting (GB) algorithm. XGBoost is based on the concept of gradient boosting, which is a method for building a predictive model by combining several weaker models into a stronger one, correcting errors made by the previous models. Decision trees serve as the base learners of the models, and the idea is to iteratively add new models with each new model attempting to correct the residual error of the previous model (T. Chen and Guestrin 2016). One of the key advantages of the model is its ability to handle very large data sets with high dimensionality.

2.4.8 Historical and Real-Time Data

Historical data plays a crucial role in many forecasting models. This data serves as the basis for identifying patterns, trends, and correlations over time. By extrapolating this data into the future, forecasts can be produced. In this project, historical data is used to train a predictive LSTM model. However, relying solely on historical data can pose some challenges. Learning patterns from past data does not necessarily mean optimal performance when faced with real-time, unforeseen circumstances. This is where the role of real-time data becomes critical.

Real-time data is often used in situations where rapid responses are necessary. Also, real-time data might be beneficial if the phenomenon being forecasted is highly volatile and influenced by real-time events. In the context of train delay prediction, real-time data poses as a crucial complement to historical data as it allows the model to adjust its predictions according to the current situation. Integration of real-time data is not without its challenges. Data processing and incorporation into a model need to go swiftly, demanding efficient data handling mechanisms.

2.5 Use of Data in Rail Industry

The fast development of computing technology and the expanding availability of data have made it possible to analyse larger and more complicated data sets, igniting interest in analytic and data science (Petropoulos et al. 2022). As a result, the forecaster's toolkit has expanded and become more sophisticated, producing remarkable outcomes. With techniques like neural networks and other varieties of machine learning, computer science has led the way, and forecasters and decision-makers are paying close attention to them. Earlier studies using various approaches are discussed in chapter 3.

Overall, the rail industry is already taking advantage of data to improve safety, reliability, and customer experience. Railway operators are using real-time data from sensors and other sources to monitor train movements, track conditions, and other factors that can affect train performance. Railway companies are using data from sensors and other sources to predict when equipment is likely to fail, allowing them to schedule maintenance activities before a failure occurs. They are also using data to optimize train schedules and routes, allowing for more efficient and effective movement of trains and reducing delays. Further, data is used to optimize energy consumption, reducing costs and environmental impact. As data collection and analysis capabilities continue to improve, we can expect to see even more innovative uses of data in the rail industry in the future.

It is well understood from literature that the quality of the data serves as an upper constraint on a machine learning model's performance (Jain et al. 2020). Poor data quality can significantly decrease the effectiveness of the related data applications (Wand and R. Y. Wang 1996). Ensuring good data quality is therefor essential. Data can be characterized as physical entities that can be stored, retrieved, elaborated upon, and sent through a network (Maguire 2007). Data quality is a multidimensional concept, consisting of the attributes accuracy, completeness, consistency, and timeliness, that can, when measured correctly, indicate the overall quality level of data. Data quality is affected by the design and production procedures used to generate the data. Understanding what quality is and how it is measured is a prerequisite for designing for better quality.

Large amounts of train operation data have been gathered and stored through modern railway signal and control systems. Nevertheless, train operation data was only retained for a short period of time to aid in the investigation of potential accidents (Wen, P. Huang, Zhongcan Li, Lessan et al. 2019b). The railway companies first started keeping their logs in the 21st century after realizing the value and necessity of this data. Advances in computer science and railway storage devices have made it possible to collect, store, and analyse real-time train operating data, making it possible to comprehend delay propagation processes from a data-driven perspective. In comparison to the current non-science-based methodologies, the genuine historical data offers more insight into the times or regularities that effect delay propagation since it is the consequence of all pertinent influencing factors.

CHAPTER

THREE

RELATED WORK

Rail operating companies have long prioritized avoiding and reducing the negative effects of delays. Numerous models and algorithms have been presented as a response to the expanding amount and availability of data in the railway sector in an effort to enhance train services. Additionally, a variety of new opportunities for operations management in rail transportation are made possible by the quick advancements in monitoring, communication, and data technology (Thaduri et al. 2015).

3.1 Train Delay Prediction Methods

Goverde and Ingo A. Hansen (2000) declared that it is possible to analyze delay propagation and conflicts using detailed information of event times associated to train services from data records of the Dutch train describer TNV-system (Telecommunication Network Voltage). The real-time record of train description has a precision of a second and can be used as a helpful tool in railway traffic control. Conte and Schöbel (2007) demonstrated how data-based approaches might be used to conduct an organized analysis of dependencies between delays. The goal of the study was to learn about the relationships between delays in order to identify their root and describe how they spread throughout the system. In the research by N. Olsson and Haugland (2004) primary causes of railway delays and the resulting comparison findings are addressed and discussed. They highlight crucial factors to take into account in advance of improving railway punctuality based on empirical findings from Norwegian research. Delays can be brought on by a variety of factors including systemic ones like signal communication and cable failure as well as human ones like the flow of passengers, occupancy ratio, and incorrect operation as well as natural events like changing weather. According to their own and other prior study, the management of boarding and alighting passengers appears to be a critical success element for punctuality on local and regional trains in congested locations. However, as stated by Sørensen, Bjelland et al. (2018), it is challenging to get actual data on train ridership. Their study looked into how mobile phone data can be used as a different source of information to analyse the number of travellers on trains and their travel patterns. Their research demonstrates that it is possible to integrate mobile data with information on train infrastructure and traffic. The primary conclusion from the research is that studies of ridership may be possible using data from mobile phones.

In the study by Berger et al. (2011), a stochastic model was put forth and tested on the German rail network to simulate train delay propagation and compute arrival and departure time distributions. Kecman and Goverde (2015) proposed a traffic state delay propagation model that estimates train running and dwell times based on the volume of traffic at the moment. Their statistical research demonstrates the relationship between dwell times and running times and the time of day and any current delays. In addition, the research showed that delays and running times rely linearly on one another for minor delays up to a certain threshold, particularly for trains with significant running time supplements.

Taleongpong et al. (2022) examines at the complex non-linear interactions among the various spatiotemporal factors that control the spread of secondary delays, which can escalate across the railway network and cause even more significant disruptions. They develop a framework that predicts key performance metrics, secondary arrival delay, secondary departure delay, dwell time, and travel time using a variety of machine learning approaches in conjunction with data mining methods. As compared to conventional delay prediction systems, the study offers frameworks for accurately forecasting KPIs using cutting-edge ML models. The study also shows that predicting the departure delay from the previous station is the most important feature between prediction of arrival and departure.

3.2 Data-driven Models

Data-driven methods for train delay prediction are single step forecasts that utilize simulation or historical system data to assess the robustness of a complex system, often producing deterministic predictions (Yin et al. 2022). Adjetey-Bahun et al. (2016) suggests a simulation-based model that measures passenger delay and passenger load as resilience indicators which allows for assessing how resilient the studied system is relative to a perturbation, by taking crisis management procedures into account. The given model incorporates all subsystems and their inter-dependencies, simulates operation circumstances for railway transportation systems, and assumes that all data are deterministic. The Parisian railway system was used to evaluate the model. The model resembles reality both during the system's regular functioning and during perturbations. The model's findings demonstrate that the system is robust since trains continue to operate there regardless of whether they decrease their speed. D. Xu et al. (2019) showed the effectiveness of their method by examining the link between inherent traits and system resilience using historical data. Using the power of data mining and machine learning techniques, they developed an advanced business analytical system to proactively predict potential disruptions and assist the operational team in improving organizational resilience. Nair et al. (2019) paper developed a large-scale, data-driven ensemble forecasting system to generate forecasts, using two-stage random forest model to increase the prediction accuracy of train recovery times. The ensemble, tested on the Deutsche Bahn, performed overall better than constituent models giving high fidelity forecasts. For the purpose of estimating conflict-free running times and dwell times, Kecman and Goverde (2015) presents a number of data-driven approaches. They discovered that decision trees and linear regression models were clearly outperformed by random forests, which provided the most precise predictions of running times.

3.2.1 Neural Network Models

Neural Network models are among the most widely used data-driven models. Recurrent neural networks, which were proposed as a solution to forecasting problems by Connor et al. (1994) , offered more accurate time-series prediction abilities than traditional neural networks. As with other neural network techniques, it was claimed that when using recurrent neural network designs, the input configuration is essential to successful prediction performance. Another pioneering studies, in 1996, was by Martinelli and Teng (1996) who developed an neural network model for optimization of solutions for the train formation problem. The study shows how the neural network model was effective in identifying the limitations of the conventional model as well as its objective functions. Martinelli and Teng offers a demonstration of how the model can generate train formation plans quickly and accurately. Yaghini et al. (2013) presented an artificial neural network model to predict train delay of passenger trains. A variety of architecture strategies and input approaches where tested, and decision trees and multinomial logistic regression models where used in order to evaluate the quality. The findings demonstrate that the delay prediction model has high accuracy and requires little training, making it a valuable tool for railway operators. Oneto et al. (2018) developed a neural network train delay prediction model for extensive rail networks. The model employs machine learning and statistical techniques to take use of big data analytic methodologies and data processing technologies through its framework. The outcomes demonstrated that advanced analytical techniques outperformed existing state-of-the-art methodologies by up to two times.

3.2.2 Fine-Tuning Deep Learning Models

Continual learning was proposed by Shon et al. (2022) as an approach to leverage pretrained deep learning models on large historical data sets followed by fine-tuning for specific tasks. Fine-tuning is the process when a pre-trained model is further trained on a new, usually smaller, data set that is different, but related to, the original data set (Too et al. 2019).

The study primarily focuses on image classification tasks, but the underlying principles and theoretical insights offers valuable guidance in refining models for other tasks. An LSTM model could, for instance, be trained on a historical train data set, and be continuously fine-tuned on real-time data. Continual learning could provide a framework for maintaining the balance between retaining past knowledge learned from the historical data, and adapting to new patterns and information provided by the real-time data.

3.3 Impact of Weather Conditions

There has been some recent research on how weather conditions affect disturbance in the railway industry, however the examination of published literature has a heavier emphasis on other modes of transportation (Nagy and Csiszár 2015; Zakeri and N. O. Olsson 2018). Yet, as Palmqvist (2019) pointed out, the literature has begun to pay more and more attention to the need to comprehend how weather and climate change affect railway delays and punctuality.

Inclusion of weather variables in the train delay prediction models have been showed to improve the overall accuracy of the models. Oneto et al. (2016) developed a dynamic train delay prediction system that uses state-of-art tools and techniques to combine heterogeneous data sources and interact with dynamically changing systems in order to provide meaningful information to traffic management and dispatching operations. The model can rapidly comprehend the knowledge that is hidden in exogenous weather data and historical train movement data. When only historical information about train movement is used, robust models with impressive performance are produced for the real train delay prediction system. These models are further enhanced by include exogenous meteorological data (Oneto et al. 2017; Oneto et al. 2016). P. Wang and Q.-p. Zhang (2019) attempted to comprehend the trends in weather and railway delays. According to the study, train delays in severe weather are most influenced by the type of poor weather, but delays in fair weather are primarily influenced by the duration and frequency of previous train delays. Further, P. Wang and Q.-p. Zhang (2019) found that infrequently occurring weather, such as snow in southern cities, has a stronger influence and causes longer train delays.

The impacts of weather phenomena including wind, temperature, and precipitation on railway operators' performance of passenger train services are estimated by Xia et al. (2013). The study was conducted in the Netherlands, and concluded that 4-8% of all train disturbances were caused by bad weather conditions, and that disturbances to infrastructure are significantly reduced by wind gusts, snow, precipitation, temperature, and leaves. Ling et al. (2018) examines the correlation between train delay and the length of time that trains are exposed to snow and rain. They discovered that trains frequently display the same delay patterns as in the past and that adverse weather frequently results in greater than usual train delays. Ludvigsen and Klæboe (2014) analyzed impact of the winter on freight rail operations in five European countries. The majority of winter freight train delays were caused by cold days, strong winds and snowfalls, which may be reasonably predicted a few days in advance. The study found that 60% of late arrivals in Finland between 2008 and 2010 were connected to winter weather by modeling the co-variation between harsh weather and freight train delays. The weather factors influence the punctuality of trains on the Norwegian railway Nordland Line was examined by Zakeri and N. Olsson (2018). The study demonstrates that winter extreme cold is a significant influencing factor that leads to delays and poor punctuality. Moreover, snow depth is the meteorological factor that most explains changes in passenger train punctuality both daily and weekly. Palmqvist, N. O. Olsson et al. (2017) investigated the impact of weather on railway punctuality. As the temperature drops below 0 degrees, punctuality reduces exponentially; at -5 degrees, it drops by 7.5%, and at -30 degrees, it drops by 50%. The variation in temperature was highly correlated with the travel distance. P. Huang, Wen et al. (2020) introduced a train delay prediction model that included a fully connected neural network with two LSTM components. The study highlighted how crucial it is to take into account the interactions between trains and stations as well as weather-related factors in terms of prediction accuracy.

3.4 Review Articles

Many review articles of train delay prediction approaches have been published recent years, which is consistent with the development of research and methodologies.

Narayanaswami and Rangaraj (2011) presents, examines, and discusses a wide range of significant railway scheduling and rescheduling activities that have been documented in the literature. The difficulties of seeing rescheduling as an operational or real-time control and scheduling as a strategic or tactical operation are further highlighted. As operational feasibility at a hierarchical level depends on the efficacy of modeling at other hierarchical levels, the dependency between scheduling and rescheduling and the influence of one on the other's effectiveness need to be investigated more thoroughly. In particular, Narayanaswami and Rangaraj emphasizes that there is a scarcity of requirements, models, and strategies for rescheduling at strategic levels and in real-time when disturbances are discovered just-in-time.

Ghofrani et al. (2018) provides an extensive overview of the areas and means that big data analytics have been used in relation to railway transportation systems. The study examined 115 papers in total. In order to spur further study on the subject, they have established a categorization framework that highlights a variety of research gaps, potential directions, and difficulties for big data analytics applications. The big data classification framework constitutes four levels, namely big data analytic domains, levels of analytics, models, and techniques, all in context of railway transportation systems.

The research by K. Y. Tiong et al. (2023a) examines prior research on train delay prediction models for data-driven methods, focusing on technical development. The study clearly emphasizes the identification of a modeling development framework with three steps, including design idea, modeling, and assessment, for train delay prediction.

Through a systematic review, the paper disaggregate the framework into six aspects: scope determination, model inputs, data quality, methodologies, model outputs, and evaluation techniques. For each aspect, the paper synthesizes important problems and techniques while discussing research gaps. The analysis shows that the majority of studies focus on creating complex algorithms for solely predicting delays at the next stop, which have limited practical applications. As stated by K. Y. Tiong et al. (2023a), the basis of comparison between different data-driven approaches is hardly established. This is due to the individual paper characteristics, where researchers tend to use their own data sets and develop specific prediction algorithms. As a result, research findings are the byproducts of a project's structural components and structure, which degrades the foundation for comparison. The paper also underlines the importance for modeling studies to prioritize data quality and conduct thorough evaluations of the model's representation power, explainability, and validity. Further, relatively few studies evaluate other elements of model performance than accuracy, making it challenging to judge the models' use in real-world applications.

In Wen, P. Huang, Zhongcan Li, Lessan et al. (2019a) paper, data-driven models for the railway dispatching problem are briefly examined, with an emphasis on relevant studies on delay distribution, delay propagation, and rescheduling. 153 papers on data-driven methods are presented and discussed. The study demonstrates how several solutions may be produced to support dispatchers' decision-making using data-driven models based on train operating records. By implementing hybrid models, which combine data-driven models with conventional mathematical models, the models may preserve the benefits of the model-driven approach and eliminate the need for precise modeling. Wen, P. Huang, Zhongcan Li, Lessan et al. asserts that machine learning or deep learning techniques have demonstrated promising potential in modeling and data processing. They may be used for train dispatching modeling and delay classification, delay propagation modeling, conflict detection and solutions, and buffer time allocation optimization.

The recent article by Spanninger et al. (2022) provides an extensive literature review on train delay prediction methods. Methods are classified and discussed in accordance with the fundamental paradigm of modeling, the precise mathematical model, the input data employed, the kind of prediction, and its horizon. The purpose of the article is to explore recent trends and research gaps, such as the impact of increased data available of railway operations on delay prediction, and to offer some guidance on which prediction model and data to employ in a given case. One of their discoveries is that, in terms of prediction accuracy, data-driven techniques, such as Neural Networks and Random Forests, may outperform event-driven ones. When utilizing data-driven models, the overfitting risk is mentioned. The power of event-driven systems, which clearly describe the dynamics and relationships of railway traffic, lies in their ability to provide predictions that can be understood, as well as their greater robustness to interruption situations. With the growing availability of data and the expected increase in railway traffic, a hybrid method that combines network modeling and big data utilization may become even more relevant in the future. The paper also discusses the issue of timetable changes in rail traffic data set. In order to successfully train and test a prediction model, Spanninger et al. (2022) clarifies that the approach must be able to abstract from these schedule changes and concentrate on invariant traffic dynamic features. This is due to the observation, stated by Oneto et al. (2018), that the delay behavior changes significantly only after a change in the nominal timetable. Further, the paper states this as a key reason restricting and limiting the time period of a rail traffic data to train prediction models. Only six of the 40 publications they analyzed used models that were trained on more than two years of data, whereas 21 studies used six months or less of observations.

3.5 Precision Metrics

To evaluate the model's performance, the majority of research just consider prediction accuracy. The precision metrics mean absolute error (MAE) and root mean square error (RMSE), both of which are further addressed in section 5.5.6, are frequently used to assess the accuracy of train delay predictions. Since MAE and RMSE are both on the same scale as the dependent variable, they are well-liked because they are simple to compare different models. Table 3.5.1 below gives an overview of the previous studies' MAE and RMSE.

The study by Corman and Kecman (2018) presents a stochastic train delay propagation model based on Bayesian networks. Bayesian networks have the ability to concisely represent the complex interdependencies between train events. The impact of the prediction horizon of the model´s prediction accuracy is analysed through MAE. The prediction horizon of 60 minutes was divided into 1 minute wide intervals and evaluated by computing the mean value of all corresponding absolute prediction errors. Figure 3.5.1 shows MAE for each considered prediction horizon. The grey band represent 1st and 3rd quartile, median and average are reported as lines, respectively in dots and solid ones. The figure clearly illustrates how MAE increases as the larger prediction horizon is considered. Longer prediction horizons result in a rise in MAE error, which shows that prediction accuracy is lower. Further, the increasing band size also indicates a lower level of confidence in the predicted values as well as increased error and higher variability of the error. There is still a considerable level of uncertainty regarding the timings of events. However, the median error of the prediction model, the dotted line, seems to increase slowly compared to the average error. This indicates that while the majority of predictions continue to be accurate, the huge variations are harder to foresee (Corman and Kecman 2018).

Zhongcan Li et al. (2022) considers the arrival routes of predicted trains and route conflicts with forward trains at multi-line stations into account when developing a train arrival delay prediction model. The study's goal was to demonstrate how accurate forecasts of train arrival delays at multi-line stations may effectively assist plans for rescheduling train operations and prevent delay spread throughout the railway network. The model was tested on historical data from the Chinese high-speed railway

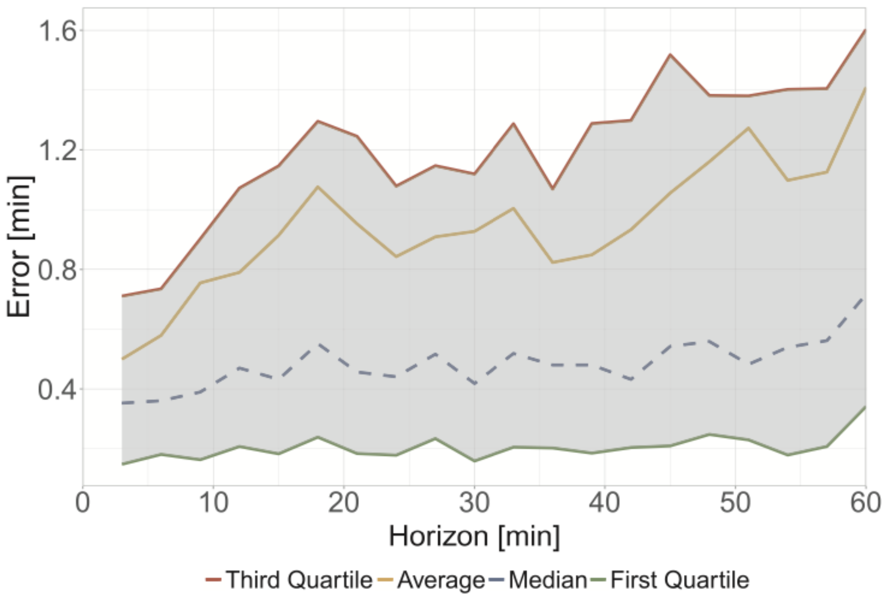


Figure 3.5.1: MAE for all considered prediction horizons, from the study performed by Corman and Kecman (2018). The band represent 1st and 3rd quartile, median (dotted) and average (solid) are reported as lines.

network and evaluated with the performance metrics.

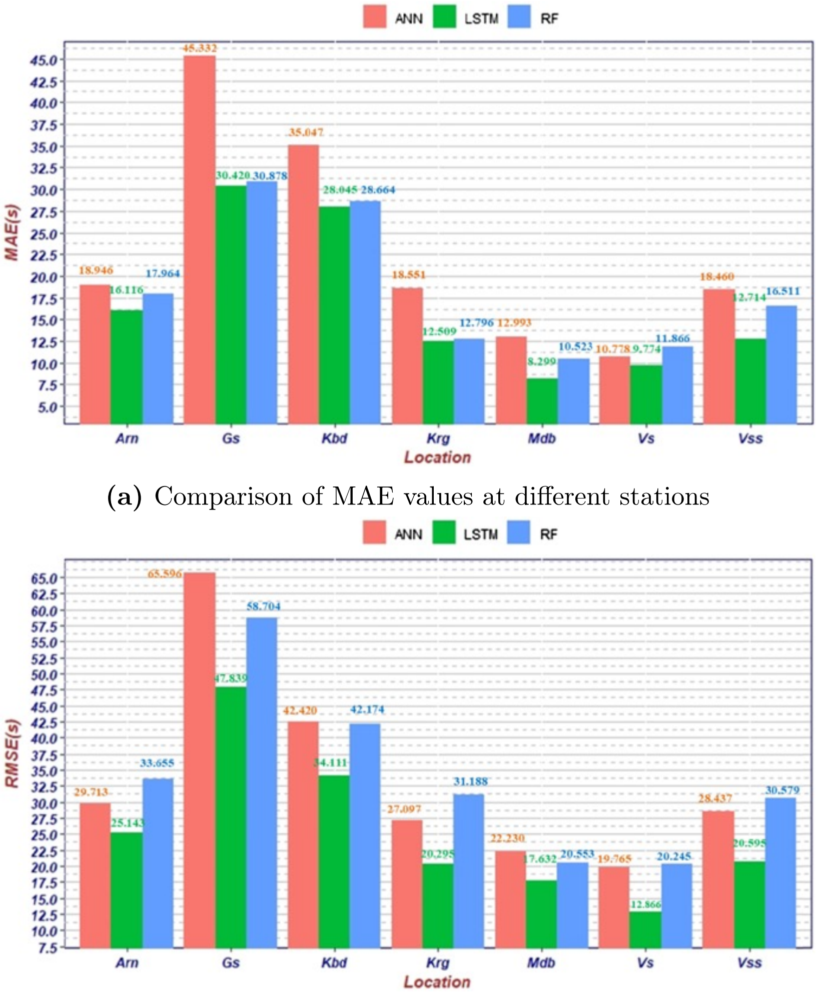
Nair et al. (2019) measures RMSE for various prediction intervals, distinct train class types, and operational statuses using their large-scale, data-driven ensemble forecasting engine. Their RMSE for all running trains was 5.52 minutes, for all urban services it was 8.45 minutes, and for all long-distance trains it was 13.08 minutes. ZhongCan Li et al. (2021) developed a near-term train delay prediction model to predict the delay 20 min later. The study separated the samples into two groups based on delays of three minutes or more and less in order to develop each group's prediction model in accordance with the Dutch national punctuality criterion. They first analysed and extracted influencing delay propagation factors, further used as model inputs. A data-driven train delay prediction model with random forest regression algorithm was then established. The findings showed that the present delays are the most significant if the delay is greater than 3 minutes. If the delay, however, is less than 3 minutes, the time during which it occurred has the largest impact on the delay 20 minutes later. This is aligned with the corresponding MAE and RMSE, which can be found in table 3.5.1. For the rule-driven automated technique of predicting train delays under several scenarios, J. Wu et al. (2022) employed a random forest implementation. The study demonstrates how crucial it is to have access to high-quality data when utilizing deep learning to forecast time series. Both MAE and RMSE were used to assess the prediction model's accuracy when combined with various baseline techniques.

Study	Model	Type	MAE (min)	RMSE (min)
Nair et al. (2019)	Data-driven	all operational trains	-	5.52
	Data-driven	all urban services	-	8.43
	Data-driven	all long-distance trains	-	13.08
ZhongCan Li et al. (2021)	RF	operational train $>$ 3min	1.708	2.863
	RF	operational train $\leq$ 3min	0.687	1.681
	ANN	operational train $>$ 3min	1.857	3.051
	ANN	operational train $\leq$ 3min	0.788	3.051
	XGBoost	operational train $>$ 3min	1.724	-
	XGBoost	operational train $\leq$ 3min	0.723	-
J. Wu et al. (2022)	RF	operational train	0.82	1.18
	CNN	operational train	0.98	1.47
	LSTM	operational train	0.66	0.78
Zhongcan Li et al. (2022)	RF	high-speed railway network	1.513	2.4
	RF	high-speed railway network, route-related variables	1.223	2.196
	DELM	high-speed railway network	2.389	3.842
	DELM	high-speed railway network, route-related variables	1.591	2.82
Skjøren (2022)	Event-driven	urban services, route R10	2.52	8.63
	Event-driven	urban services, route R11	2.29	4.92
	Event-driven	urban services, route L12	2.4	10.2
Sørskår (2022)	RNN	one long-distance train	1.23	1.73
	LSTM	one long-distance train	1.17	1.69
	GRU	one long-distance train	1.17	1.68

Table 3.5.1: Overview of other literature's using MAE and RMSE to evaluate the performance of their proposed model.

List of abbreviations used in table 3.5.1:
ANN = Artificial Neural Network, CNN = Convolutional Neural Network, FCNN = Fully Connected Neural Network, RNN = Recurrent Neural Network, LSTM = Long Short-Term Memory Networks, GRU = Gated Recurrent Unit, RF = Random Forest, XGBoost = eXtreme gradient boosting, DELM = The Deep Extreme Learning Machine

Wen, P. Huang, Zhongcan Li and Mou (2019) used a Long Short-Term Memory (LSTM) train delay prediction model to examine the correlation between different railway system features and train arrival delays. The model outperformed the random forest model and the artificial neural network model in comparison to accuracy tests of prediction accuracy. The performance metrics were measured and compared for LSTM, RF and ANN models of different stations. The result was presented in histograms, shown here in figure 3.5.2a and figure 3.5.2b.



(a) Comparison of MAE values at different stations

(b) Comparison of RMSE values at different stations

Figure 3.5.2: Comparison of MAE and RMSE values at different stations from the study performed by Wen, P. Huang, Zhongcan Li and Mou (2019)

3.6 Specialisation Projects

Both the authors of this Master's thesis completed their own specialisation project in the Autumn semester of 2022. The project *Predicting Train Departure Delays*, written by Sørskår (2022), developed a train delay prediction model based on historical train data. The project *Evaluation of Train Delay Estimates*, written by Skjøren (2022), examined the performance of Bane NOR's existing train delay prediction model. Upon completion, it became evident that the projects complemented each other. Thus, the work conducted during these projects served as a groundwork for this master's thesis to build upon.

The specialisation report written by Sørskår (2022) explored the utilisation of various machine learning models with the purpose of predicting train departure delays. The study found that recurrent neural network models were well suited for train delay prediction, with the best performing models yielding Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) of 1.17 min and 1.69 min, respectively. In contrast to this project, the study only examined one single train going in one direction along the line Oslo S through Trondheim, reducing the complexity of the task. Additionally, the study underlined the importance of including past delay patterns in form of lagged observations, or *time lag*, of the target variable. A significant improvement in predictive accuracy were evident upon inclusion of these *time lags* features. Finally, the study found that the results were comparable to results from previous, similar studies.

The specialisation project written by Skjøren (2022) examines the performance of Bane NOR´s train delay prediction model on three Norwegian railway routes. The three routes examined where all running through the capital area, Oslo area, but only in one direction. The study illustrates how trains estimated departure times are predicted to be before their actual departure from a station. This illustrates that the train delay prediction model is too optimistic. The results show an overall high estimate deviation median for all three routes, above 40 seconds. The study also illustrates how rarely a departure time is negative, estimated to be after the actual departure. Several stations experiences high estimate deviations, but it is hard to see any clear pattern between the estimate deviation and the actual delay. On the other hand, a clear pattern through the day is to be seen, where the time periods with more delays are affected by bigger and more distributed estimate deviations. Bane NOR´s train delay model performance varies between the three routes, examined through both correlation analysis and performance metrics.

CHAPTER

FOUR

DATA

Chapter 4 presents the data used and its limitations. In this study, three data sets are gathered from two separate data sources.

Rail data two data sets with rail traffic data, provided by Bane NOR.

Data set 1 consists of raw data of traffic logs, utilized in the analysis phase. The gathered data is from the period between 15.06.22-22.03.23. For the analysis of the current train delay prediction model, data-logs from TIOS records have been collected consecutively and saved. The main objective of the data was to investigate the precision of the current used train delay prediction model.

Data set 2 consists of pre-processed traffic data in the period from 01.01.21. to 28.02.23. The other data-set is also TIOS records, but is excludes the estimated arrival and departure records. This data-set is going to be subject to various feature engineering techniques for the purpose of employment and training of a new prediction model.

Weather data one data set with historical daily meteorological observations have been gathered from the Norwegian Meteorological Institute

Data set 3 weather data in the period from 01.01.21. to 28.02.23.

To distinguish between the two rail data sets, the succeeding material will refer to data set 1 as the *TIOS raw data set*, and data set 2 as the *TIOS pre-processed data set*.

As stated by Spanninger et al. (2022), timetable changes in historical traffic data can be a challenge when training and testing a model, further commented in section 3.4. TIOS raw data have a time period on 10 month, and the TIOS pre-processed data spans on two years and two months. The restricted period prevents delay behavior changes in response to adjustments in the nominal timetable. Further, the data, their purpose and limitations are further described below. Further information about the different methodologies can be found in chapter 5.

4.1 Rail Data

Rail traffic data is historical data of train movements, collected from the national database TIOS. TIOS receives route data and dynamic train movements from the remote control systems, as well as information about train composition from the railway undertakings. There are historical records for every train, including the operating company, the class, and the train's number, at every location. Additionally, TIOS receives GPS data from the running trains. Figure 4.1.1 illustrates the data flow for each scenario.

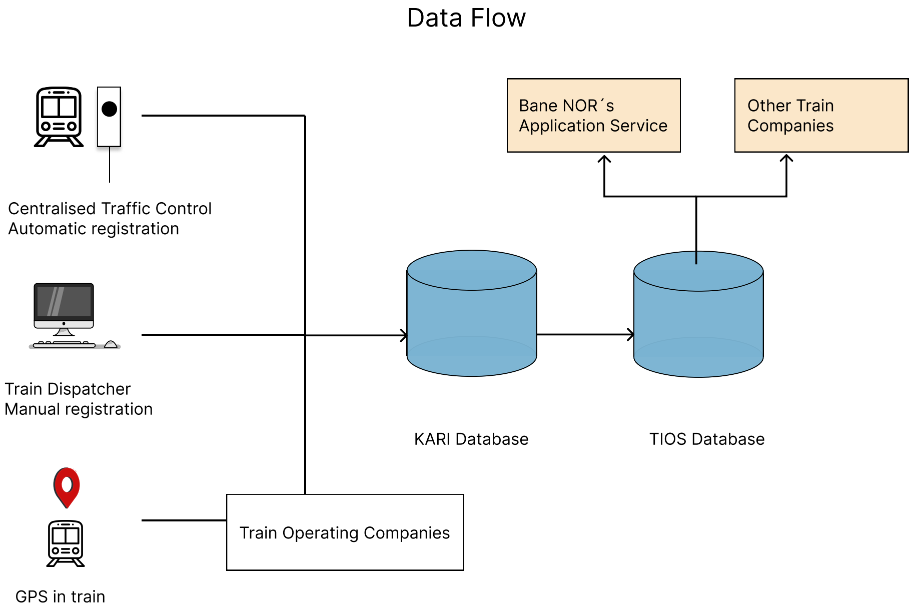

Data Flow
Bane NOR´s Application Service
Other Train Companies
Centralised Traffic Control Automatic registration
Train Dispatcher Manual registration
KARI Database
TIOS Database
GPS in train
Train Operating Companies

Figure 4.1.1: Illustration of the data flow of the train record information.

When a message is read into TIOS, a time stamp with the date and time is created for each record. Both route data and real-time are included in the data collection. Route data is predetermined information such a schedule, stops, fares, network, framework, and general details. Route data complies with the NeTEx standard, which establishes the data structure and service descriptions for the transmission of line information, timetables, and other pertinent data for public transportation (Statens vegvesen 2019).

The remote control system on the railways allows for the automated registration of real-time data as well as deviation data in the database known as KARI. Data is manually registered on a few railway routes in Norway, including Gjøvikbanen and Nordlandsbanen. However, the line chosen in this study automatically saves data. Continuous modifications and deviation data that are relevant to the current day's operations are included in real-time data. KARI data adheres to the European SIRI standard, which specifies minor alterations to the scheduled timetable. SIRI messages can be separated into three main categories (Vegdirektoratet 2019), presented in table

4.1.1 below.

Tag	Description
SIRI-ET	Estimated Timetable, defines updated estimated times
SIRI-VM	Vehicle Monitoring, updates of positions, as well as real-time delays are communicated here
SIRI-SX	Situation Exchange, updates if any deviations

Table 4.1.1: Real-time, SIRI messages used in this project

Every time it passes a sensor for a remote control system, SIRI-ET and SIRI-VM are continually transmitted, updating real-time information for the railway network. Both NeTEx and SIRI are XML (Extensible Markup Language) formats that make it easier to share real-time information on public transportation as well as route data between distant systems (Bane NOR 2022).

Arrivals and departures information is sent to the TIOS database. This project is looking at the TIOS departure logs. From the perspective of the passenger, the time dimension is deemed to be greater in importance. The most recent status for a train's departure is provided to passengers using today's traffic information systems. Train arrivals are also announced, but they are not as readily available as departures. This is most likely a result of passengers being more concerned with the train's departure time than its arrival time. Every end-station, however, will only have an arrival time and no departure time. The end-stations in both directions are left out of the constructed model since it only focuses on departures from stations in this research.

Different tags are used in the TIOS logs for various message types. The TIOS log *STATION*, where both scheduled arrival and departure are logged, contains a pre-defined timetable. The estimated and actual records, both from the real-time data given by KARI database, are the other TIOS logs used in this project. *ETD* (Estimated Time of Departure) and *ATD* (Actual Time of Departure) tags are used to identify departure data. Table 4.1.2 below lists the tags used in this project along with explanations of each tag and its message.

As shown in table 4.1.2, both scheduled departure time and scheduled arrival time are part of the STATION TIOS log. The pre-scheduled timetables departures will be referred to as STD (Scheduled Time Departure). Additionally, *city* is a location code that is placed at both passenger stations, where passengers board and depart trains, and *base stations*, which are locations where trains only passes by.

Estimate records are regularly generated as a train moves through the network and the system receives new data about a train's actual position. Therefor, TIOS receives many estimates for each station, especially when a train is delayed early in the route. However, only the latest estimate for each station is stored over time in TIOS. When evaluating the current prediction model, all estimates throughout a route are needed. Therefor, the two different data sets have been gathered for this project.

Tag	Description	Nested tag
All tags	TIOS messages with nested XML syntax. All logs includes the nested tags where they constitutes the combined primary key	origin_time, city, train_number, *
STATION	Scheduled timetable. A message for each station on the train's route, including train number, location code, scheduled arrival and scheduled departure.	scheduled_arrival, scheduled_departure
STD	Scheduled time departure. A part of the STATION messages for the scheduled departure.	scheduled_departure
ATD	Actual time departure, update message at a station and gives the actual departure of the train.	actual_departue
ETD	Estimated time departure, forecast data for new estimated departure values for delayed trains.	estimated_departue

Table 4.1.2: TIOS tags and their description, used in this project

4.1.1 Case: Oslo - Trondheim

The chosen railway line studied in this master travels from Oslo to Trondheim. The train route constitutes both *Dovrebanen* and *Gardemobanen*. Dovrebanen stretches from Trondheim to Eidsvoll which constitutes 485 km of railroad, mainly single track. Gardermobanen stretches from Eidsvoll and Oslo which constitutes 66 km of railroad, mainly double track. Today, it operates as an important national route between Trøndelag and Eastern Norway. The line consists of 67 stations, where 20 of them are stop-stations, and the remaining 47 stops are intermediate stops. An illustration of the route can be found in figure 2.2.1 marked in orange and the corresponding line map can be found in appendix A.

There are a total of 12 train rides between each endpoint, six in each direction, every day. Two of the trains rides at night, and they are excluded in this study. Route tables for both directions can be found in Appendix B. The train numbers can be used to determine the time of day, where it increases throughout the day. Morning trains have the smallest train numbers, while evening trains have the largest train numbers.

4.1.2 Limitations

Timetables are pre-scheduled times for arrivals and departures for train routes in the order of minutes. When the designated minute is nearing and the driver is given permission to begin moving, the train can leave a station. However, time registrations are often based on automated registrations from signal systems with a resolution measured and recorded to the second. This indicates that there is a good likelihood that the scheduled time log will be somewhat different from the actual and estimated time logs.

Measurements from the actual remote signal system installed on the rail network are recorded in TIOS real-time traffic logs. The readings are thought to be precise timers for when trains enter and leave specific locations. The registration points, however, are not precisely where trains stop at a station, which might provide measurement-related difficulties. In order to account for the "typical" time the train will take to go from the registration point to the train's actual stop, an offset is often employed at stations. Likewise applies for departures. This implies that there is no genuine precision even when actual times are measured in seconds. The accuracy of the real-time recordings is more unclear the more away a sensor is positioned from the station.

The measurements, however, are same across all trains since they use the same methodology and measuring position. Real-time information will differ between the stations and may not necessarily represent how customers perceive the rail traffic (N. Olsson, Halse et al. 2015). When comparing actual and estimated time records from TIOS, there is nothing incorrect with the measurement itself because every train is measured the same at every stop, at the same location. There will be a difference between the real signal records and the scheduled time records, nevertheless.

4.2 Weather Data

Historical weather data were obtained from the Norwegian Meteorological Institute (MET) in the period from 01.01.2021 to 28.02.2023 (Institute 2023a). Initially, access to MET Norway's archive of historical weather and climate data was gained through its Application Programming Interface (API). The Frost API provides access to quality controlled daily, monthly and yearly measurements of various weather elements such as precipitation, temperature and wind data, along with metadata about weather stations.

In order to retrieve relevant weather data, the locations of the weather stations of interest had to be collected. Through usage of the additional API, Google Maps API (Google 2023), coordinates were drawn from the train stations along the railway line. These coordinates were further used to retrieve the most nearby corresponding weather station number.

Further, the actual weather element data from these stations were then collected, retrieved by using the command: $geometry = nearest(POINT(latitude, longitude))$. This command outputs the station ID for the station closest to the given coordinates, along with other metadata. Setting the station IDs of interest, along with the weather elements and the time period to be included, will result in the retrieval of JSON ready for analysis.

The downside of using the API for data retrieval is the less organized format of the JSON file compared to a fully formatted .csv file. Further, the content of the JSON file had to be verified as there is no automatic verification as to whether the desired data content was included or not. The laborious analysis of the data obtained through the API revealed that there were actually significant gaps in the data for both the requested station IDs and the relevant time period. The data availability is visualized in appendix D, which was plotted by using the same coordinates obtained from the Google Maps API.

As an alternative, MET Norway also provides a web based portal, seklima.met.no (Institute 2023b), providing access to the same database through a more user-friendly interface. The data availability is here plotted on the Norwegian map, providing weather statistics for specified weather stations in Norway. From this portal, the time period and the weather elements were chosen and cherry-picked, before an automatic compilation of a ready-to-use .csv file containing the weather data.

Data column	**Description**
NAME	Name of the Weather station
STATION	Station ID - SN****
DATE	DD.MM.YYYY covering the time period for each station and for each weather element
VALUE	Daily weather element value.

Table 4.2.1: Weather data retrieved from seklima.met.no

The following weather elements were chosen and gathered for this analysis:

Temperature Daily mean temperature. An arithmetic average of 24 hourly values.

Precipitation Daily sum of precipitation

Snow depth Daily registration of snow depth - snow coverage measured in centimeter above ground.

Wind Daily mean wind value. An arithmetic average of 24-hourly values.

4.2.1 Limitations

There exist several limitations with regards to the weather data. Firstly, and most importantly, the availability of data for both the desired time period and weather elements varies greatly between the different weather stations. As shown in the visual representation in appendix D, there are several gaps in the data coverage along the railway line, particularly for snow depth data. For some stations, data is only available for a few months, while for others, data is available for the entire time period. This can be due to various reasons such as malfunctioning equipment or insufficient funding for maintenance and repairs. Although the visual representation of data availability in the web portal which were ultimately preferred to the API gave a more comprehensive data collection, this method also yielded gaps especially in the chosen time periods.

Furthermore, the process of retrieving weather data through APIs can be time-consuming and cumbersome, particularly when dealing with JSON files. This can also lead to the possibility of errors in data analysis and interpretation.

It is also worth noting that the weather data provided by MET Norway is subject to quality control, but there may still be errors or inaccuracies in the data due to various factors such as measurement errors or station location changes.

CHAPTER

FIVE

METHODOLOGY

The study has employed a variety of methodologies. This chapter introduces and describes the different methods and tools employed.

5.1 Literature Review Method

A literature review was done to gather information and analyse the complexity of the railway system, the gap in delay forecasts, and the status of current knowledge. In the work of the specialization project articles, during Autumn 2022, the first round with a literature review was formed. Both authors delivered their respective specialisation project; Sørskår (2022) and Skjøren (2022). The literature reviews in both specialization project papers served as the foundation for the literature review in this master's thesis. In addition, a comprehensive review of the literature was conducted to supplement the existing review, cover new theoretical areas, and convey the most recent findings. This master's thesis employed the same research methodology that was developed and applied for the article for the specialization project. A comprehensive description of the methodology follows below.

From the start of the study, a broad review was conducted before it was narrowed down to provide a systematic perspective. The papers from conferences, academic books, and manuals were examined. Annual reports from rail operation businesses and recent research reports from research institutes have also been examined. Only papers published in English and Norwegian proceedings as of 2010 were included. However, the reviewed material on the history of railway systems and its research subject was evaluated regardless of when it was written. The reviewed material on the history of Artificial Intelligence (AI) consists of textbooks and articles written by recognized authorities in the field. Most of them are widely used in academic settings and are considered seminal work in the AI field. The completed literature review produced results and information that are quite intriguing and useful for this report.

Google Scholar, ScienceDirect, Scopus, and Oria were the digital databases used. Along with terms that encompass the subjects relevant to each phase, keywords that pertain to railway transportation, such as "rail", "railway," and "train," were employed. Five stages can be taken to break down the conducted literature search.

Step 1 examined the foundation of the railway system and its difficulties in light of train delays and the operational planning stage. The handbook by Profillidis (2006) was studied, especially the chapters "Railways and Transport", "Policy and Legislation", "Forecast of Rail Demand" and "Planning and Management of Railways". During the literature search, the terms "delay," "real-time," "dispatching," and "prediction" were employed. Based on the findings from the first step, the subsequent steps supplement with more information.

Step 2 examined on how tactical planning's pre-work helps avoid delays and additional propagation. The terms "timetable," "robustness," "buffer time," and "rescheduling" were employed.

Step 3 examined the use of and open data policy in the rail sector. Research reports and literature reviews were both done. In addition, both domestic and foreign open data information pages were examined.

Step 4 examined the evolution of accurate real-time train delay prediction, the models used and its advancement regarding to the increased amount of data. "Big data", "data-driven models", and "delay prediction models" were employed as keywords.

Step 5 examined machine learning models and their use in earlier studies. The terms "Feature Engineering", "Recurrent Neural Network", and "Machine Learning" were employed in this step.

Each phase employed a combination of keywords to search through titles, abstracts, and literature keywords to ensure the quality and applicability of the evaluated papers. Papers from the mentioned literature were also reviewed. Papers without full-text availability or of purely low quality were not included. A literature review was conducted for each of the specialization project. Further literature study were established in the early phase of the master. The entire amount of material reviewed and included in this study, given its broad reach, consisted of 187 distinct articles.

5.2 Data Management

Building understanding of the data and its provenance is the goal of the data management process. Data management is the process of gathering, arranging, and gaining access to data to improve decision-making and increase productivity. A data management procedure aids in ensuring that data is accurate, accessible, and secure. The exploratory data analysis identifies and resolves any potential issues with data. This data insight is essential for the pre-processing of the data. Data wrangling is the process of transforming raw data to facilitate subsequent analysis. It includes preparing data for analysis by cleaning, manipulating, and integrating it. Data wrangling aims to enhance the quality of the data and make it more appropriate for the data mining process. Exploratory data analysis and data wrangling constitutes the infinite loop of data science.

Depending on the data source, we often have different expectations for quality. The three different data sets were all received in very different conditions and therefor also required different scale of data wrangling and exploratory data analysis. This indicates that it was necessary to introduce three separate data management processes, one for each individual data set, including both data wrangling and exploratory data analysis.

5.2.1 Data Wrangling

Data wrangling has been defined as "a generic phrase capturing the range of tasks involved in preparing your data for analysis" (Rattenbury et al. 2017). In this stage, the data quality was examined, files were formatted, structure was simplified, and new measurements were generated for further use in the analysis. Each data-set was separately examined due to their condition and further purpose.

5.2.1.1 TIOS raw data, traffic logs with train records

The data set with TIOS-logs was received as 814 files, each formatted as .txt format. Each file consists of raw data of TIOS logs in form of list of strings. Each log has a timestamp for when the record was received, and the train record itself. The TIOS train records are XML formatted, a nested structure with a set of information. The table below shows how the data logs were before and after the first data wrangling phase.

Before: [dd.mm.yyyy hh:mm:ss]: <TIOS-tag ><nested-tag>...</nested-tag></TIOS-tag>

After: <TIOS-tag><nested-tag>...</nested-tag></TIOS-tag>

First phase of data wrangling focused on structuring and formatting the files. Each file looped through the list of strings. The timestamp, ([dd.mm.yyyy hh:mm:ss]:), was first separated from the rest of the string. By using the Python module *BeautifulSoup*, each TIOS log message on xml-format was located by its TIOS tag. For the desired TIOS tag *STATION*, *ATD* and *ETD*, logs were obtained and stored in the corresponding lists. The ETD TIOS record's timestamp was saved for later examination, while the timestamp was taken out of the other two tags. Table 4.1.2 provides a summary of the tags that are being used, their descriptions, and nested tags. Further, the parser *"lxml"* extracted the desired data from the xml-strings. The strings with TIOS xml-messages were now contained in the three separate lists, each of which represented a TIOS tag.

Before: <TIOS-tag><nested-tag>...</nested-tag></TIOS-tag>

After: TIOS-tag Dataframe:

nested-tag 1	nested-tag 2	...	nested-tag n-1	nested-tag n
value 1_1	value 1_2	...	value 1_{n-1}	value 1_n
value 2_1	value 2_2	...	value 2_{n-1}	value 2_n
...	...	...	...	...
value $n-1_1$	value $n-1_2$	...	value $n-1_{n-1}$	value $n-1_n$
value n_1	value n_2	...	value n_{n-1}	value n_n

Second phase of the data wrangling was further structuring of the data. TIOS messages were converted from lists of strings into rows of tables. Three tables, also known as dataframes in Python, represented one TIOS tag each. Dataframes with their rectangular, regular structure are used in this project because they are easy manipulate and analyse. The converting process also conducted a set of cleanings. The nested TIOS-tags that were not needed in this project was removed. Further, the train numbers that are not used in this project were also cleaned out. All time record were converted from a string to a Python´s datetime object, as it is easier to manipulate when working with date and time. Finally, missing values and NaN (not a number) were removed from each dataframe. The three final dataframes serve as the basis for additional calculations and analysis of the TIOS data logs, further explained in chapter 5.4.

Three dataframes were now representing the scheduled, actual and estimated data. Scheduled is represented in the dataframe STATION. The dataframes could then being merged for further analysis of the data. A main key was created before merging the related dataframes.

Key: train number, origin time and station code

The values of the nested-tags *train number*, *origin time* and *station code* together make up a unique train record identifier, hence they are utilized as the combined primary key in all dataframe merging operations. The key locates and gathers the correct logs before merging the dataframes.

For the analyse of the TIOS traffic logs, section 5.4, further calculations were made. See table 4.1.2 in chapter 4 for TIOS-tag descriptions. The following equations, 5.1, 5.2 and 5.3, were introduced.
The time difference between the actual departure time and the scheduled departure time is used to calculate the actual train delay (AD).

$$AD = ATD - STD \tag{5.1}$$

The difference between the corresponding actual estimated departure time is used to calculate the estimate deviation (ED).

$$ED = ATD - ETD \tag{5.2}$$

The difference between the associated predicted departure time and the scheduled departure time is used to calculate absolute estimate deviation (AED).

$$AED = ETD - STD \tag{5.3}$$

As earlier mentioned, timestamps were saved and appended to each corresponding ETD train record. The timestamps are used to delineate which estimation that should be taken into consideration in the analyse. TIOS database receives estimated arrival and departure time that are predicted continuously throughout a route. The estimates are predicted both very early before and close to an actual arrival and departure. This is because the train delay prediction model used by Bane NOR generates predictions for the upcoming stations far in advance of their arrival. By excluding both too early and too late generated estimates, the two models are more comparable and a more reflective analyse can be established. Estimates predicted more than 50 minutes before the actual departure are eliminated. Additionally, estimates predicted five minutes or less before the actual departure are also eliminated. The longest travel between two stations, between Heimdal and Støren, is about 38 minutes, and our model's delay constraint is 10 minutes, thus the time limit is set at 50 minutes. When the correlation and accuracy of the two models are examined and compared, this constraint increases their comparability.

5.2.1.2 TIOS pre-processed data, traffic data

Since the second TIOS data set was received in a pre-processed Excel file, the extent of required re-formatting and re-structuring was minimal. However, certain portions of the data had to be removed before using it for further analysis and training of the machine learning model. Firstly, this includes detecting and removing outliers in the data. This procedure is described in section 5.2.2.1. Secondly, delay records from base stations had to be removed from our data set before training the prediction model. This procedure is described in section 5.3. Common for both are presence of bias in our data, which is highly undesirable when training a machine learning model.

5.2.1.3 Weather data

As described in the Data section, the desired weather elements were cherry-picked for the relevant station along the railway line. The results were a pre-processed .csv file containing the weather data. However, one major limitation of using weather data from MET Norway is the varying degree of data availability for different weather elements.

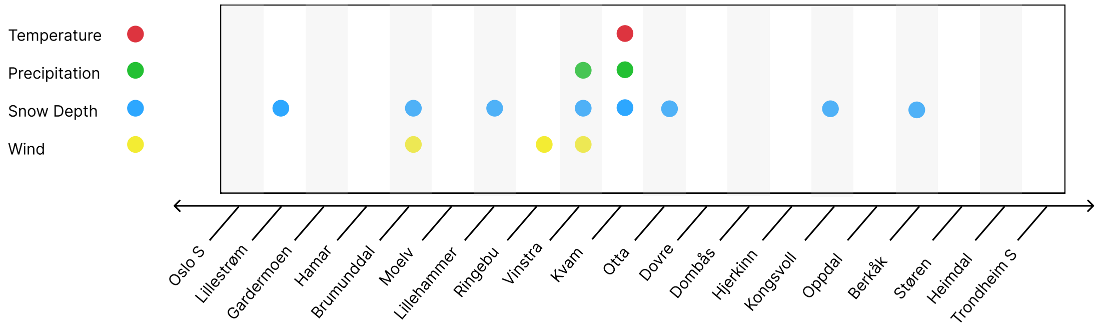


Figure 5.2.1: Data availability for four weather elements at all stop-stations along railway line Oslo S - Trondheim. Dots indicates the missing data.

Figure 5.2.1 contains information about the missing elements in the weather data. Dots indicate missing values for the given weather element at close proximity of the given train station. Missing information in the data set can make processing and analysis of the data more complicated, and it can introduce a degree of bias in the data. One solution to this is list-wise elimination of cases with missing values. Another, better alternative is the process of *data imputation*. Data imputation is defined as the procedure of using alternative values in place of missing data (Simplilearn 2023). There exist various data imputation techniques. Some of these include K Nearest Neighbors, Average or Linear interpolation, etc. However, for time series data, there are specific techniques who are more efficient and simpler than others. Time series data have a sorted structure wherein nearby values are more comparable than far-off ones. Therefore, the next or previous value inside the time series, in this case the value at the next or previous station, has been substituted for the missing value. In the case of Snow Depth at Kvam station, an average was calculated from the previous and next station.

5.2.2 Exploratory Data Analysis

Exploratory data analysis (EDA) was introduced by John Tukey as a different approach to data analysis that deviated from the conventional use of confidence intervals, hypothesis testing, and modelling. "Exploratory data analysis is actively incisive, rather than passively descriptive, with real emphasis on the discovery of the unexpected," (Tukey 1972) according to Tukey, who views it as a philosophical approach to managing data.

Exploratory Data Analysis is the process of examining and analysing a data set by utilizing various statistical and graphical techniques to better understand the relationships and patterns within it. While summary statistics provides a summary of the data sets central tendency and variation, will data visualization identify patterns, trends, and clusters in the data. Both techniques have been used.

The three data sets were separately explored through EDA to get better insight into and knowledge about their data properties. Key properties, like structure, granularity, how fine/coarse is each datum, scope and temporality, how is the data situated in time, are all important insight for both the data cleaning and further analysis. A thorough understanding of the data enables to determine if a problem identified is minor and can be overlooked or fixed, or whether it seriously restricts the utility of the data. EDA was therefore preformed from an early stage of the analysis and in parallel to the data wrangling process in every stage of the data lifecycle.

5.2.2.1 Outlier detection and removal

Some use-cases of machine learning include fraud-detection, medical diagnostics, etc. In such cases, the goal is to detect *anomalies* in the data. An anomaly is a data point which do not fit the normal pattern of the data. The goal of this project is to create a prediction model which can predict "normal", or "everyday-delays", and not large, unpredictable delays resulting from unforeseen events. In that sense, fitting the model to the normal pattern of the data is essential.

The boundary values for outlier removal were calculated using box plots. A box plot is used to show the distribution of the quantitative variables in the descriptive statistics process (Galarnyk 2022). Box plots provide a clear and concise representation of the distribution of the data, which can help in identifying the central tendency, dispersion, skewness, and presence of outliers in the data. The technique is further described and illustrated in the graphic inspection chapter 5.4.1.1.

The interquartile range (IQR) is a statistical measure of the spread of a data set. It is calculated as the difference between the third (Q3) and the first quartile (Q1) of the data distribution, which is the range between the 25th and 75th percentiles of the data. The IQR method for identifying outliers is used by setting up a "fence" outside of Q1 and Q3, or minimum and maximum values. Values outside of these boundaries are considered outliers.

$$lower\_boundary = Q1 - 1.5 \cdot IQR \approx -5.075 \tag{5.4}$$

$$upper\_boundary = Q3 + 1.5 \cdot IQR \approx 10.005 \tag{5.5}$$

Equations 5.4 and 5.5 uses Q1, Q3 and IQR to calculate the lower and upper fence around the data, or the *outlier boundary values* of the data. The equations shows that the lower and upper boundary value is set to -5.075 and 10.005 respectively.

A histogram is used to illustrate the frequency distribution of the delays against the number of occurrences in the variable category. Its simplicity and clarity make it a very useful graph. Histograms are further described and illustrated in the graphic inspection chapter 5.4.1.2. Figure 5.2.2 shows the data distribution before and after outlier removal using the upper and lower boundary values calculated using the IQR method.

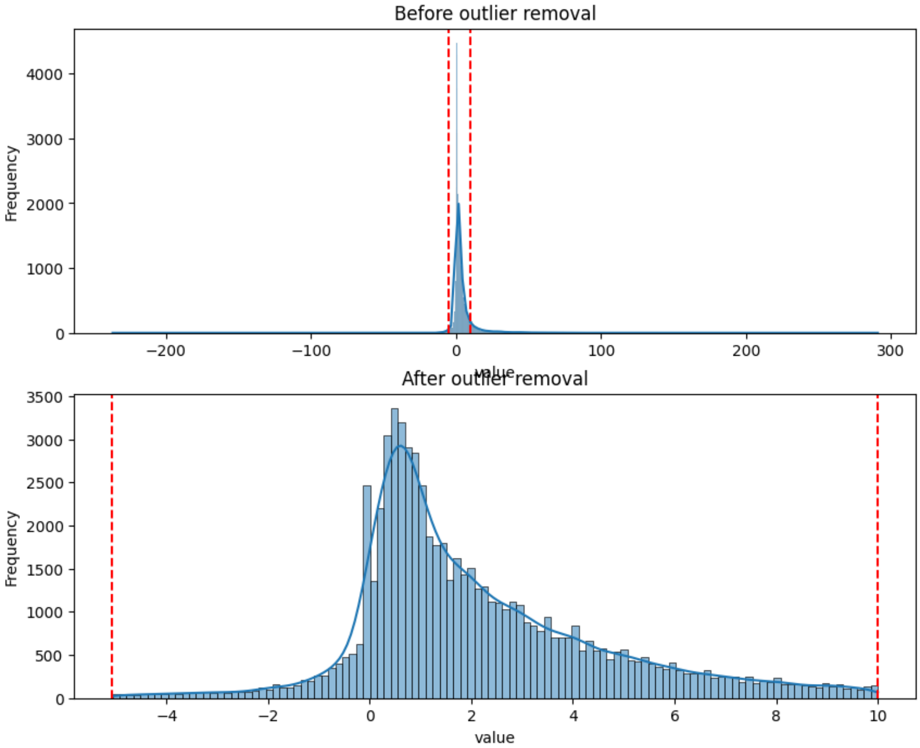


Figure 5.2.2: Data distribution before and after outlier removal.

5.2.3 Data Size

In terms of predicting train delays, the data size available for this study provides a strong basis compared to other research articles. Having a well-sized sample is crucial for prediction models because it ensures that the model's estimates are reliable and accurate. If the sample size is too small, the model may not capture the full range of variability in the data, and it can lead to overfitting or underfitting. Overfitting occurs when the model is too complex and fits the training data too closely, resulting in poor generalization to new data. Underfitting, on the other hand, happens when the model is too simple and fails to capture the underlying patterns in the data. Therefore, having a sufficient sample size is essential for building robust, accurate, and reliable prediction models. For a comprehensive review of data-driven approaches used to forecast train delays, including their scope, methodology, and data size considerations, please refer to the study by K. Y. Tiong et al. (2023a).

The three data sets all have different time span and data size, presented below:

TIOS raw data, traffic logs with train records, the total of data received consisted of 814 files with the total of 251.554.101 train records. After the data wrangling process, the data size constituted 2.715.816 train records.

TIOS pre-processed data, traffic data, the total of data received was pre-processed, consisting of 328 675 train records.

Weather data, daily weather data were gathered from the Norwegian Meteorological Institute, for four different weather elements:

Temperature A total of 13 673 data records.

Precipitation A total of 14 202 data records.

Snow depth A total of 9 468 data records.

Wind A total of 12 624 data records.

Due to the varying availability of data for the different elements, data imputation methods were utilized to get a complete data set. This procedure is described in the Analysis section.

5.3 Feature Engineering

Feature engineering is the process of transforming raw data into features that can be used to improve the performance of machine learning algorithms. A feature is a variable used to describe some aspect of a data object (G. Dong and H. Liu 2018). E.g., in the case of time series data describing a railway line and its occupying trains, a feature could be time-and-space related, or related to the trains' physical attributes. The quality of the features can have a significant impact on the accuracy of the models. G. Dong and H. Liu (2018) defines feature engineering broadly as a workflow that could include several phases such as feature transformation, feature generation, feature selection, feature analysis and evaluation. When performing feature engineering techniques on your data, domain knowledge is essential in order to effectively extract meaningful features from the raw data.

Domain knowledge has been gained from the TIOS pre-processed data through an EDA. The data set spans from January 1st 2021 to February 28th 2023, and the data was received pre-processed and formatted in a .csv file containing 328675 rows and 13 columns.

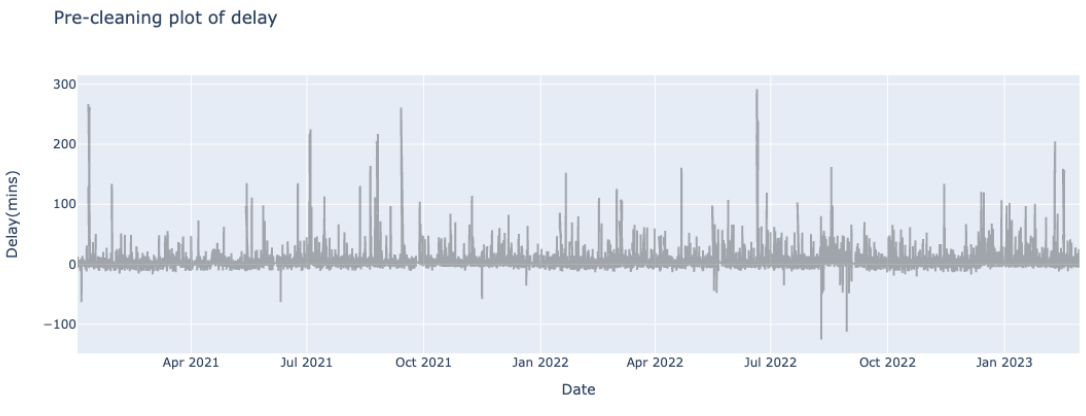


Figure 5.3.1: Plotting target variable, departure delay, before cleaning the data.

Variable	**Description**
Year	The year the train journey took place
Date	The date the train journey took place
Train_num	The identification number of the train
Sequence	The order of the train in a particular journey
Station	The name of the train station
Station_code	The code associated with the train station
Planned_arr	The planned arrival time at the station
Actual_arr	The actual arrival time at the station
Planned_dep	The planned departure time from the station
Actual_dep	The actual departure time from the station
Train_set	The identification number of the train set
Arr_delay	The delay in arrival time
Value	The value to be predicted by the model, called the target variable, the delay in departure time

Table 5.3.1: Received data set format

The problem at hand is predicting the departure delay at every stop-station, i.e. stations where passengers can board and exit the train, along the railway line from Oslo to Trondheim. The line consists of 67 stations, where 20 of them are passenger stations, and the remaining 47 stops are base stations. An immediate thought was to exclude the base stations from the data set, as predicting delays at these locations is not of interest. As one can observe from 5.3.1, there exists negative departure delays in the data set. This means that sometimes trains leave the train station before the scheduled time, which is a major inconvenience for passengers. After analysing the data, a total of 44 332 occurrences of negative departure delays were found. However, only 7 723 of these corresponded to delays at stop-stations. In practice, a negative delay at a base station could mean bigger dwell times at stop-stations, which again means higher probability of departing the stop-stations without any delays. Therefore, the departure delay at the preceding base station at a given stop-station should be used to inform the model about possibilities of future departure delay. A new feature is created, delay_grad, and if the delay at the preceding base station is:

$$\forall Delays = \begin{cases} < -5 minutes & \text{delay\_grad} = 1 \\ \geq -5 and < 1 minutes & \text{delay\_grad} = 2 \\ \geq 1 and \leq 5 minutes & \text{delay\_grad} = 3 \\ > 5 minutes & \text{delay\_grad} = 4 \end{cases}$$

This feature, known as 'delay_grad', was included at a later stage of the modelling process, as it was hypothesized that additional information about past delays could significantly affect the predictions. For clarity and enhanced comprehension throughout this thesis, this term will henceforth be referred to as *delay classification*. After this feature is extracted from the departure delay at base stations, the delays are excluded from the data set. This is due to the bias they create in the data, as it is not preferable for the model to potentially predict negative delays at stop-stations. This means a big reduction in data size, but it preserves the integrity of the data that should be used to train the model.

Before starting the feature transformation phase including deriving new, useful features out of existing ones, the model needs to be informed that there are trains going in both directions, north- and southbound. It is sufficient to create a binary variable indicating the direction. Next, the "station" column is encoded into numerical values, and columns which are deemed statistically insignificant are discarded. The Python library Pandas and its functions set_index() and to_datetime() is used to set the "Date" column as DatetimeIndex with assigned Timestamps.

Additionally, all features should be converted into the correct data types enabling the model to understand the data it is being fed. This includes converting strings from the data set into integer or float types. Also, the DatetimeIndex objects can be used to extract features such as "day_of_week" and "week_of_year" using built-in functions from the Pandas library. Additionally, index.date can be used to extract data about holidays, which could potentially result in higher demand and thus increase possibilities for delays. Finally, index.hour is transformed into cyclical hour_sine and hour_cosine time features to let the model know that 23:59 is close to 00:01, etc.

A final, popular feature engineering technique is employed - adding lagged observations, or *time-lags* of the target variable to the feature space. The idea is that there is often patterns or trends in the time series data that can be used to make predictions of future values. Looking at past values of the target variable can help the model in identifying these patterns and use them to make informed forecasts. Finding an optimal number of these time-lags to include as features could help optimize the predictive performance of the model. The number of time-lags to include is highly dependent on the nature of the data and the model, hence finding a standard which count for all models is infeasible. The effect of including these values will be further analysed when performing the correlation analysis. As time lags and auto-correlation are closely related concepts in time series analysis, the correlation coefficient will likely be large. What is interesting, however, is to analyse and assess the coefficient as the time-lag window increases, and to assess the impact of increasing that window. This was done in the specialization report by Sørskår (2022), which found that the optimal time-lag window size in this particular predictive task is set to 5.

The station-to-integer mapping used on the train station column is also used to encode the corresponding weather stations in the weather data retrieved from MET Norway. The weather data should be merged with the train data to include the weather elements in the feature space. This is done using the Pandas merge() function on one Dataframe containing the train data, and one Dataframe containing the weather data. The Dataframes are joined on the "Date" and "Station" columns. The original train data set is preserved as a copy, so that the effect of including the weather elements can be further analysed through correlation analysis. This method will be further discussed in the Analysis section.

Figure 5.3.2 depicts the reduction of rows as a result of various procedures during the data wrangling, EDA and feature engineering phases. The data was received at step 0. Step 1 included removing base stations in between stop stations, as described in this section. Step 2 included removing NaN values from the merged weather data. Finally, step 3 included removing outliers from the data using the outlier boundary values described in section 5.2.2.1.

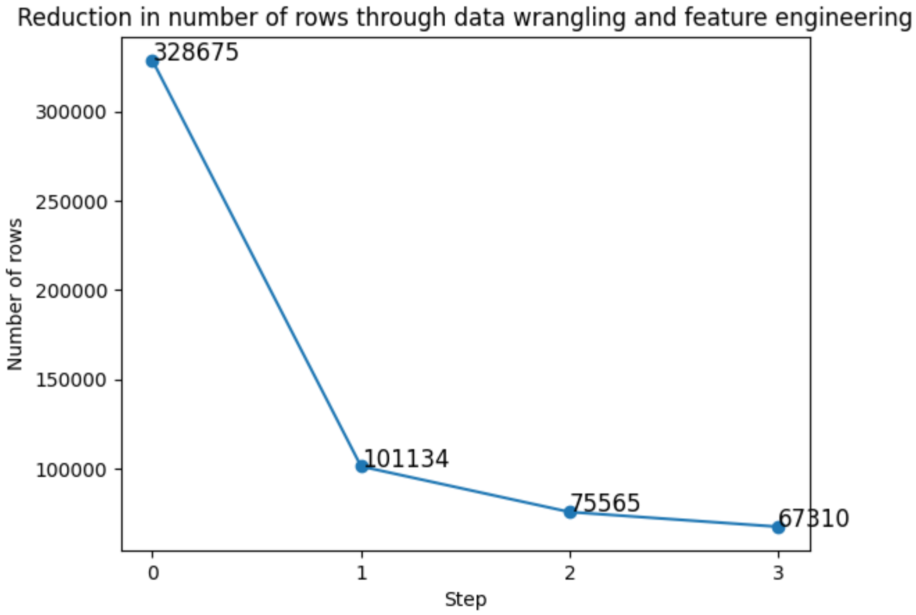


Figure 5.3.2: Reduction of rows through data wrangling, EDA and feature engineering

After performing EDA, and feature engineering techniques, the data has been transformed from a 2-dimensional Excel table to a 2-dimensional Dataframe with the following shapes: $(328675, 13) \rightarrow (67310, 18)$. The final data set and its feature space is described in table 5.3.2.

It is important to note that the delay classification feature was introduced into the feature space at a later stage of the modelling process. This was done to scrutinize its effect on the overall predictive performance. Consequently, two different models were built and trained: the LSTM model, which does not include the delay classification feature, and the $\overline{LSTM}$ model, which includes the delay classification feature.

Variable	**Description**
value	The target variable, departure delay
Delay_grad	Delay classification. Categorical variable indicating degree of delay from the previous base station
Train_num	The id of the train
Station	The id of the train station
Direction	A binary variable indicating which direction the train is moving in
Temp	The daily mean temperature in degrees Celsius
Prec	The daily sum of precipitation in mm
Snow	A daily measurement of snow depth in cm above ground
Wind	The daily mean wind speed in m/s
Lag1 - Lag5	The values of the target variable from the previous 5 time steps
Day_of_week	The day of the week
Week_of_year	The week of the year
Hour_sine, hour_cosine	The sine and cosine of the hour of the day
Is_holiday	A binary variable indicating whether the day is a holiday

Table 5.3.2: Data set after performing feature engineering

5.4 Analysis

Several analyses have been performed in order to comprehend the data. The procedures, methods, and tools are described in the following sections. Section 5.2.1 describes how the data set was prepared to the analyse phase.

5.4.1 Graphical Analysis

Graphical analysis was used to better visualize and understand the train records data and the quantitative variables. Different graphical tools depict different data features such as trend, frequency, dispersion, and distribution shape.

5.4.1.1 Box Plot

Box plot is used to show the distribution of the quantitative variables in the descriptive statics process (Galarnyk 2022).

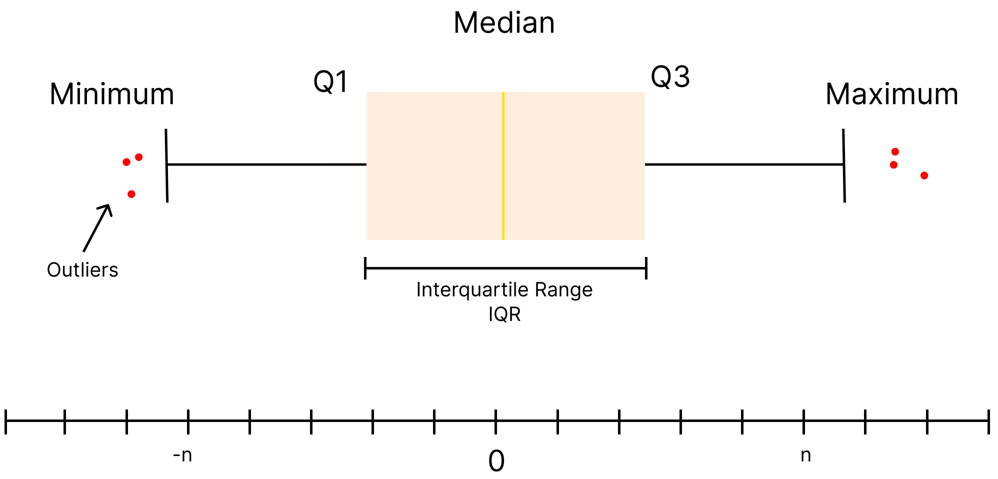


Figure 5.4.1: Box plot figure which demonstrates the minimum, maximum, first quartile Q1, third quartile Q3, median and outliers

The box plot's graphical appearance conveys the following characteristics of the data distribution:

Median, the middle value of the data set.

First Quartile, $Q1 \div 25th$ percentile, the middle number between the smallest number, not *minimum*, and the median of the data set.

Thrid Quartile, $Q3 \div 75th$ percentile, the middle value between the median and the highest value, not *maximum*, of the dataset.

Interquartile Range, $25th$ to $75th$ percentile.

Whisker, the two horizontal lines out of the box, representing the scores outside the middle $50th$ percentile, i.e. the lower $25th$ percentile and the upper $25th$ percentile of the scores.

Outliers, values that is numerically distant from the rest of the data.

Maximum, $Q3 + 1.5 \cdot IQR$, the highest score excluding outliers.

Minimum, $Q1 - 1.5 \cdot IQR$, the lowest score excluding outliers.

The different components of the box plot are displayed in figure 5.4.1. A visual representation of the data distribution is provided by box plots, which split the data into sections with bottom and upper whiskers, first and third quartiles each containing around the $25th$ percentile of the data set. Box plots show the location, spread, and skewness groups and provide brief explanations of the sample and data measurements. The box plots were created using the Python Library module Plotly since it prohibits interactive data display and enables both graphical and descriptive statistics.

5.4.1.2 Histogram

A histogram is a graph that displays the values of a numeric variable's distribution as a collection of bars (J. Chen 2023). A variety of classes are divided into columns along the horizontal x-axis in the bar graph representation of the data. The numerical count or percentage of occurrences for each column in the data are shown on the vertical y-axis. Usually, each bar represents a range of numerical values. Each bar's height represents the proportion of data points that have values that fall into a certain bin.

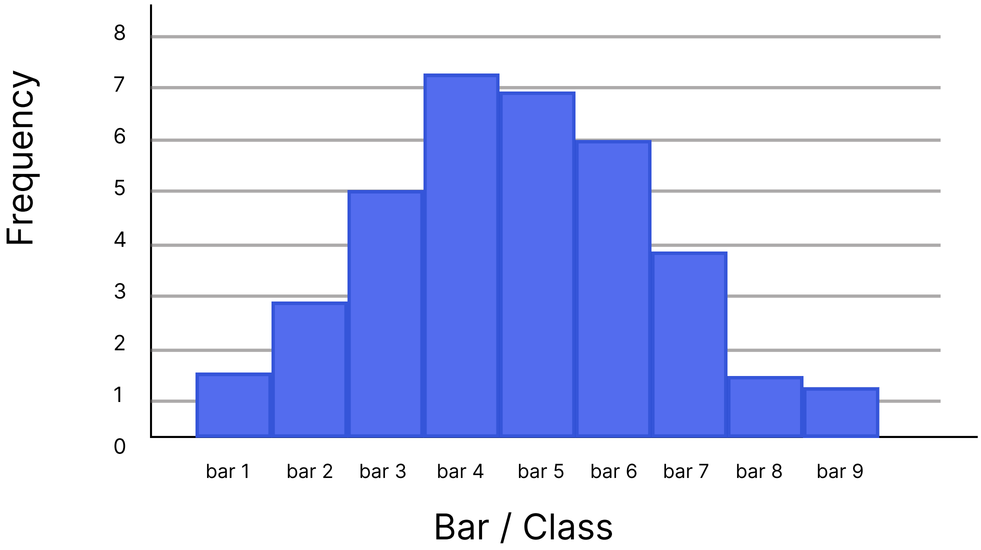


Figure 5.4.2: A histogram in which values from a collection of data are separated into bars of the same length as their frequency.

5.4.2 Correlation Analysis

Correlation analysis is a statistical technique that is used to determine the strength and direction of the relationship between two or more variables. Correlation analysis is used to understand the distributional characteristics of train operations for both the current used train delay prediction model and our developed model. By analysing the relationships between the quantitative measurements actual delay and absolute estimate deviation, the correlation strength is measured for each model using a numerical value called the correlation coefficient. This allows gaining insights into the distributional characteristics of train operations on the route and compare the results.

Significant relationships between measurements and estimations are shown by strong correlations. Finding the directions and strengths of the link might also help future study to focus its results. A correlation analysis can have one of three outcomes:

Positive correlation, if an increase in one variable leads to an increase in the other.

Negative correlation, if an increase in one variable leads to an decrease in the other.

No correlation, there is no relationship between the two variables.

Pearson correlation coefficient is used to measure the correlation because of the linear relationship between the two quantitative variables. There is a linear relationship between two variables if a change in one is proportionate to a change in the other. Equation 5.6 is a formula for calculating the Pearson correlation coefficient r:

$$r = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2(y_i - \overline{y})^2}} \tag{5.6}$$

A Pearson coefficient value of 1 represents a perfect positive relationship, -1 a perfect negative relationship, and 0 indicates the absence of a relationship between variables. The interpretations of the relationships strength, commonly referred to as the effect size, reveal the practical significance of the outcome. Correlation is an effective size and the most common way of determining linear correlation. The guide by Evans (1996) suggests the following evaluation for the absolute value of $r \in$:

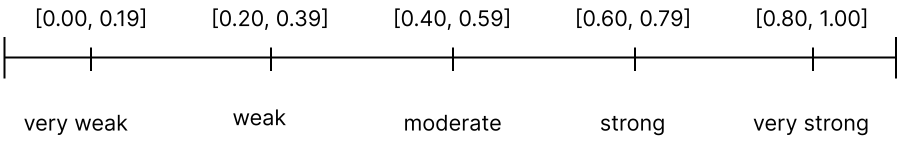


Figure 5.4.3: Evaluation of absolute value of the Pearson Correlation Coefficient.

5.5 Prediction Methodology

Departure delays at $Station_{t+n}$ is predicted based on time lags from the n previous stations, as illustrated in figure 5.5.1, in addition to other features. After completing the initial EDA and feature engineering techniques, the last step of the data preparation process is to scale and split the data. Subsequently, the model architecture needs to be defined and built in Python. Then the model has to be trained on the prepared data set, before techniques for evaluating the model performance is described.

5.5.1 Data Scaling and Split

It is common that the features of a data set contain data with on different scales. Column A could for instance contain integers valued up to 1000, while column B could contain floats valued up to 1.5. This difference in scale can cause disarray during training of the model, potentially resulting in inaccurate predictions. To mitigate this issue, all the data is scaled to help normalize the data. In this case, all features are scaled using the MinMax Scaler into a range between 0 and 1 (Bisong 2019). This benefits the model particularly in two ways. Firstly, the model avoids large weight update during training making the model converge more smoothly. Secondly, it improves generalization enabling the model to learn underlying patterns in the data more efficiently.

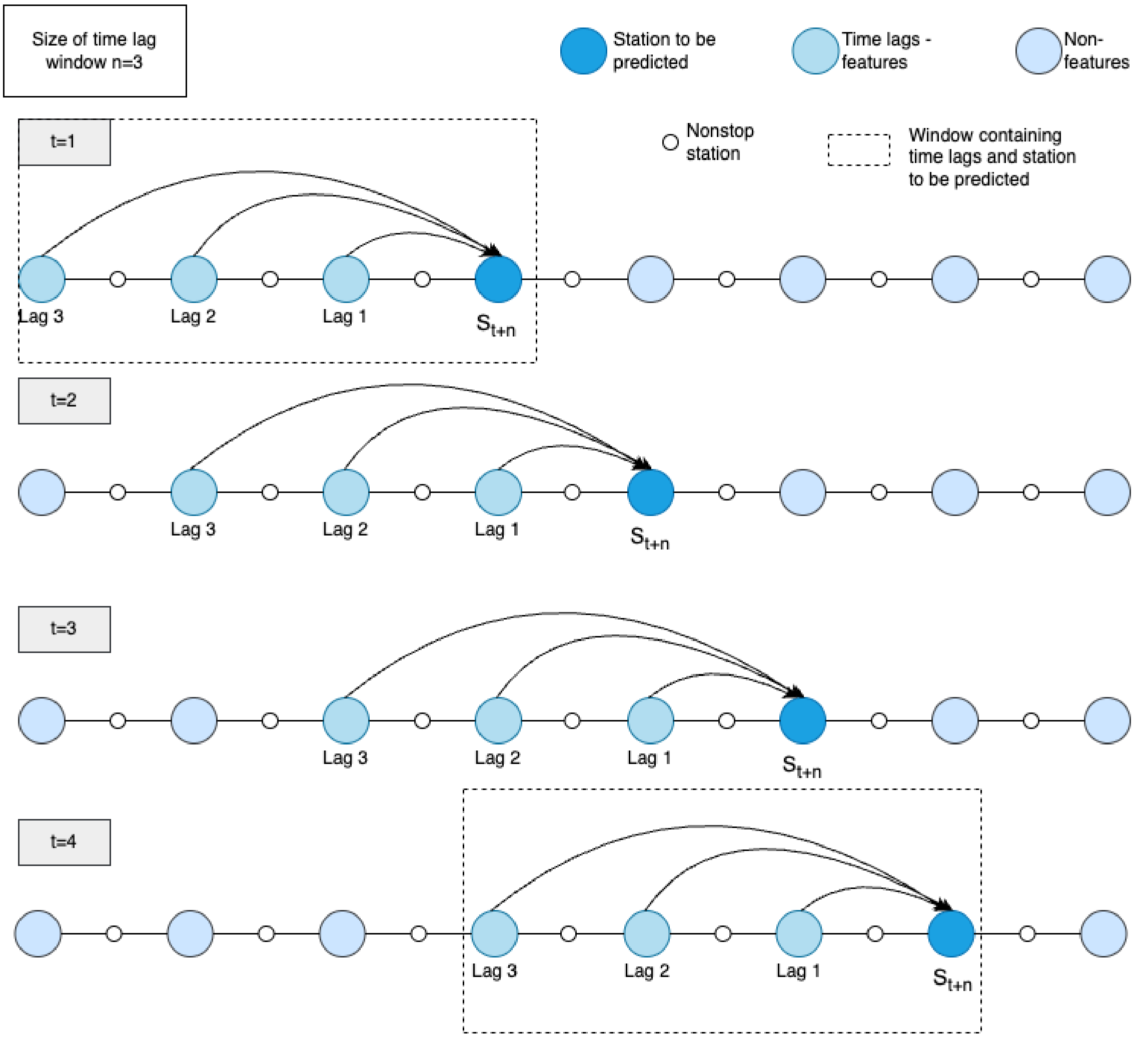


Figure 5.5.1: Using time-lags to predict departure delay at $Station_{t+n}$ with a window size of n=3, at t=1,2,3,4 timesteps.

The data is typically split into three sets: training set, validation set and test set, with each serving its own purpose. During training, the model's trainable parameters are updated based on the error between the predicted output and the actual target value. When the data is split, it creates two vectors for each set; X and Y, where X contains all variables, or features, which the model is going to base its predictions on, and Y contains the target variable, in this case the departure delay. The model tries to approximate a function f on input X such that

$$f(X) \approx Y \tag{5.7}$$

Minimizing the error on the training set alone can lead to overfitting, which means performing well on the training set alone but poorly on new, unseen data. Therefore, the model is evaluated on the validation set during training, to assess the generalization performance of the model, which is crucial for accurate predictions post-training. After training is completed, the loss from both the training set and the validation set is typically compared in a loss curve. Also, the test data is fed into the model in order to generate predictions and evaluate the performance.

5.5.2 Modelling

In this thesis, a LSTM model is tasked with predicting departure delay at stop stations along the railway line Oslo - Trondheim. The model was built using the Python library PyTorch. The model consists of three LSTM layers and one fully-connected layer from the torch.nn module, which provides building blocks for machine learning models. Table 5.5.1 and figure 5.5.2 shows the architecture and parameters of the model.

Parameter	**Value**
Input dimension	20
Output dimension	1
Number of layers	3
Number of hidden units	64
Dropout probability	0.2
Learning rate	0.0001
Weight decay	0.000001
Number of epochs	50
Optimiser	Adam
Loss function	MSE
Activation function	Tanh

Table 5.5.1: Model architecture and parameters

The parameters are explained in further detail in appendix F. The LSTM layers all have 64 hidden units, or *neurons*. The first LSTM layer takes the data as input, and performs the mathematical calculations on the input data, illustrated in figure 2.4.3 and further explained in appendix F. The data is then propagated through the entire network. The fully-connected layer, or the *output layer* is tasked with taking the output of the last LSTM layer as input, and produce a one-dimensional output, or the *final prediction* of the model.

5.5.3 Training the Model

The training procedure can start once the model architecture and optimization parameters are defined. During training, the model is fed batches of training data, and these are forward-propagated through the model network. The loss function, mean squared error (MSE,) is used to calculate the error between the predicted output and the actual value:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.8}$$

Where y_i is the actual value at timestep i and $\hat{y}_i$ is the predicted value at timestep i. The goal of the training process is to minimize the loss function. This is done using an optimisation algorithm. In this case, the Adam optimizer is chosen, due to its fast and efficient convergence during training. The optimizer updates the model's parameters, the weights and biases connecting the neurons, in the direction of the negative gradient of the loss function with respect to the parameters. The parameters are updated using backpropagation through time (BPTT). Once the parameters are updated and the error has been minimized, the next iteration of the training process begins. This involves feeding the next batch of data through the network, computing the loss, and performing backpropagation to update the parameters again. This process is repeated until the specified number of epochs is reached, or the validation loss stops improving. The training procedure and the various parameters are explained in further detail in appendix F. Once the training is complete, the final trained model can be used for making predictions on the unseen test data.

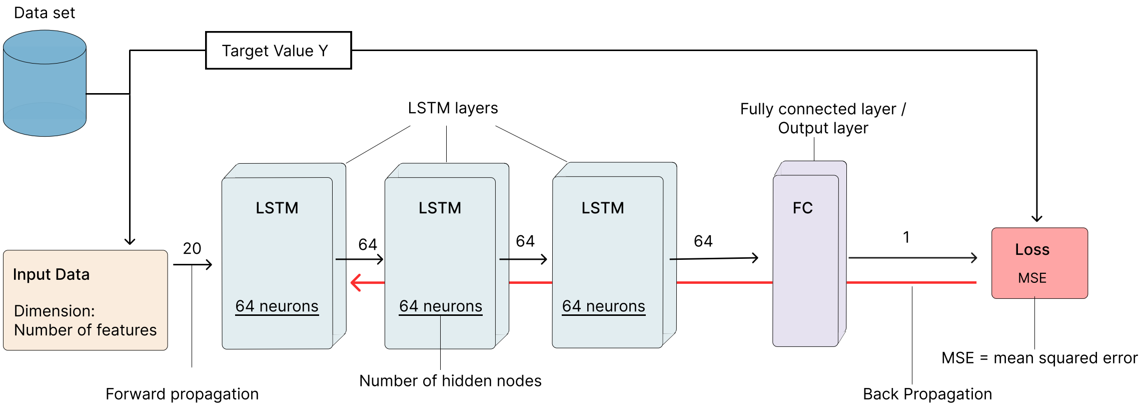


Figure 5.5.2: LSTM model architecture

5.5.4 Hyper-parameter Tuning and Evaluation

Deep neural networks like LSTMs depend on a wide range of hyper-parameters to be chosen wisely for optimal performance. Some of these parameters include choices about the network architecture, the input data, regularization and optimization (Feurer and Hutter 2019), as shown in table 5.5.2. According to L. Yang and Shami (2020), selecting the optimal hyper-parameter configuration has a direct impact on the model's performance.

There could be many possible factors causing sub-optimal prediction perform-

Parameter	**Category**
Number of LSTM layers	Network architecture
Number of hidden units	Network architecture
Activation function	Network architecture
Dropout probability	Regularization
Weight decay	Regularization
Learning rate	Optimization
Optimiser	Optimization
Loss function	Optimization
Number of epochs	Optimization
Data-split configuration	Input data

Table 5.5.2: Hyper-parameter categories

ance. The hyper-parameters involving the network architecture define the depth and complexity of the model and its ability to capture complex patterns in the data. Optimization-related parameters include the learning rate, the optimiser algorithm and the loss function. If the parameters are not well configured, the model could fail to converge to a good solution. The data-split configuration, i.e. the portion of the data which is used for training, validation and testing, is also important. If, for instance, the training portion is too low, the model could struggle with capturing sufficient amount of patterns in the data, and the performance would be poor when evaluated on the test data. On the other hand, if the training portion is too big, the model could overfit on the training data and fail to generalize, struggling with performing inference on new data. If regularization parameters are properly defined they could help prevent overfitting. As previously stated, a well composed set of hyper-parameters could easily improve the performance of the model.

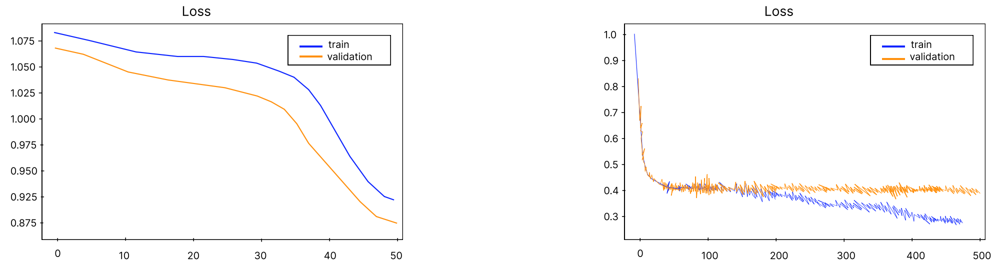
One common method for hyper-parameter tuning is the grid search technique (Bergstra and Yoshua Bengio 2012). Grid search is a systematic approach that involves searching through a predefined set of hyper-parameter values, evaluating the performance of each combination, and selecting the best-performing set of values. By performing an exhaustive search, grid search can help identify optimal hyper-parameters that lead to improved model performance. However, this technique can be computationally expensive, particularly when dealing with a large number of hyper-parameters and a wide range of values. To mitigate this, a more efficient alternative such as random search or Bayesian optimization can be employed. In this study, grid search was adopted for hyper-parameter tuning due to its comprehensiveness and simplicity. The chosen hyper-parameters and their respective ranges are listed in table 5.5.3. During the grid search process, the model undergoes training with multiple configurations of these hyper-parameters, and their effectiveness is assessed by evaluating model performance on the test data. Based on the evaluation metrics MAE and RMSE, the best-performing hyper-parameter combination will be selected and used for the final model.

Hyper-parameter	**Value Ranges**
Learning rate	0.001, 0.0001
Number of LSTM layers	2, 3, 4
Number of hidden units	64, 128, 256
Number of epochs	20, 30, 40, 50
Dropout probability	0.1, 0.2, 0.3
Data-split	80-10-10, 70-15-15

Table 5.5.3: Grid search hyper-parameter setup

A common practice for train-val-test splitting, as suggested by Goodfellow et al. (2016), is to use approximately 60-80 percent of the data for training, 10-20 percent for validation, and 10-20 percent for testing. However, these proportions can be adjusted depending on the size and characteristics of the data set. The proportions which were deemed optimal in this particular case were 70-15-15 percent for the training-, validation- and test set respectively.

To ensure accurate future predictions, it's vital to evaluate the model's ability to generalize to new data. Performance metrics, such as Mean Average Error and Root Mean Squared Error, will be explained in further detail in section 5.5.6. Another common way of diagnosing the performance, is through learning curves. A learning curve is a plot of model learning performance over time, or epochs of training, typically both for training- and validation loss. Examples of learning curves demonstrating underfitting and overfitting of the data is shown in figure 5.5.3.



(a) Underfitting the data

(b) Overfitting the data

Figure 5.5.3: Learning curves illustrating underfitting and overfitting of the data

5.5.5 Bane NOR´s Train Delay Prediction Model

The train delay prediction model that are being used on the Norwegian railways today is part of the KARI-system, developed by an external German company called *Funkwerk*. KARI-system has a goal to ensure that passengers nationwide are provided with visual and acoustic real-time information on rail services (Group 2023). Like all the other functions, the train delay prediction model is subdivided into an independent software subsystem and mapped centrally via a user-friendly interface. An illustration of the data flow in Funkwerk´s system can be found in figure 4.1.1.

The train delay prediction model from Bane NOR uses an event-driven methodology to clearly record the relationships between train events. The model is created upon a system of linear equations, with an iterative process and multi-step forecasts, to predict train delays. The ability to foresee impending railway accidents has a direct bearing on how long delays will last. The objective is to capture the processes and constraints for the dynamics of rail operation.

The estimates are being generated based on either GPS information from the train every 10 seconds, information from sensor trigger points, or manual information registration by the train dispatcher or the train driver. The following stations are automatically updated when an estimate at one station is updated, including the lead-in time between and at the stations. The estimate is made right away, however the client information is not displayed before the delay exceeds 1 minute 59 seconds. Additionally, the system uses a rounding requirement, where it don't show seconds on the customer's display, making 5 minutes and 45 seconds seem as 5 minutes. This only applies to client information, not to data in the TIOS database.

The model's delay predictions begin with the route plan and Operation Control Points (OCP). Each OCP has the following attributes:

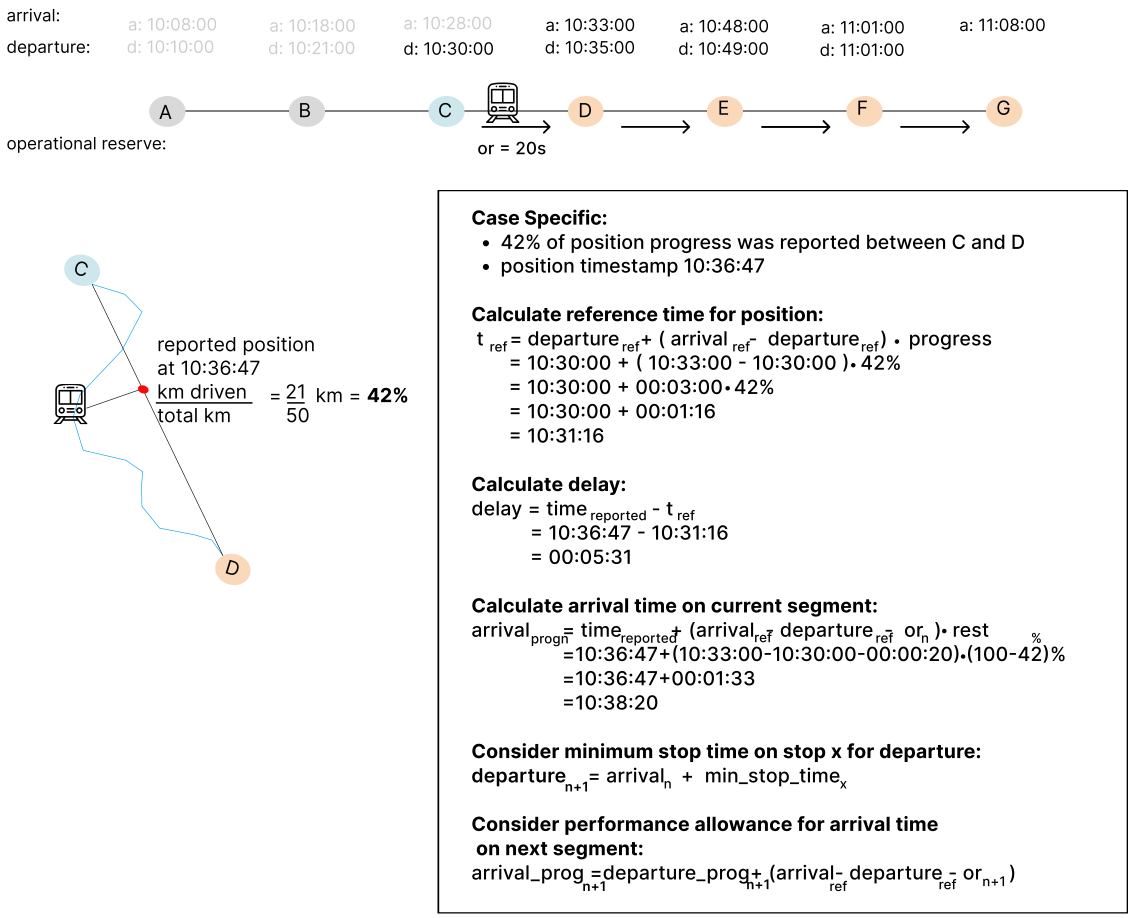
STA/STD (Scheduled Time of Arrival/Departure): These are the pre-defined arrival and departure times for each OCP based on the route plan.

runTimes minimalTime : This indicates the minimum time the train can travel between two OCPs.

operationalReserve : This represents the "slack" time or the extra time available between the planned departure time and the next arrival, considering the minimum travel time between OCPs.

stopTimes minimalTime : If the train stops at an OCP, this value indicates the minimum time the train should remain stationary.

Figure 5.5.4 illustrates a simplified specific case on how the model by Funkwerk calculates further train delays on the route. The model uses predefined parameters and Operational Control Points (OCPs). OCPs are specific points along the route plan that are used to represent necessary driving patterns. The general basis for the predictions is calculated similarly for all locations. However, when GPS data is available, the model can provide a more accurate prediction for the nearest OCP, which doesn't necessarily correspond to a station.



Case Specific:

- 42% of position progress was reported between C and D
- position timestamp 10:36:47

Calculate reference time for position:

t_{ref} = $departure_{ref}$ + ($arrival_{ref}$ - $departure_{ref}$) • progress
= 10:30:00 + (10:33:00 - 10:30:00)• 42%
= 10:30:00 + 00:03:00•42%
= 10:30:00 + 00:01:16
= 10:31:16

Calculate delay:

delay = $time_{reported}$ - t_{ref}
= 10:36:47 - 10:31:16
= 00:05:31

Calculate arrival time on current segment:

$arrival_{progn}$= $time_{reported}$+ ($arrival_{ref}$- $departure_{ref}$- or_n)• $rest_{\%}$
=10:36:47+(10:33:00-10:30:00-00:00:20)•(100-42)%
=10:36:47+00:01:33
=10:38:20

Consider minimum stop time on stop x for departure:

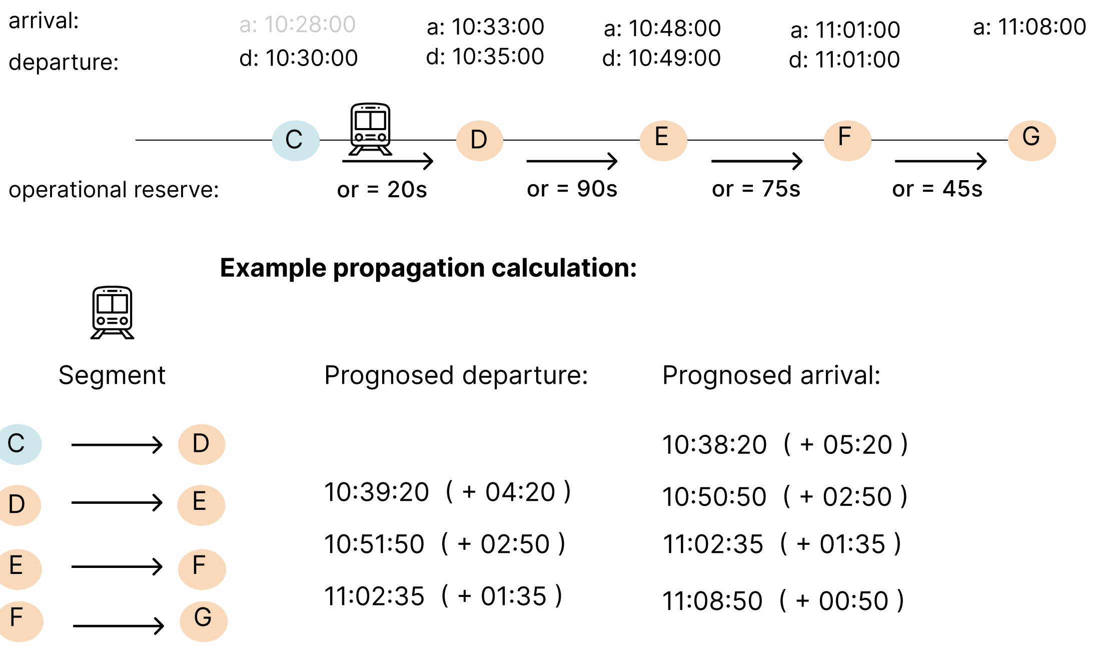
$departure_{n+1}$= $arrival_n$ + $min\_stop\_time_x$

Consider performance allowance for arrival time on next segment:

$arrival\_prog_{n+1}$=$departure\_prog_{n+1}$+ ($arrival_{ref}$- $departure_{ref}$- or_{n+1})

Figure 5.5.4: Specific case illustrating how the train delay prediction model is calculating further delays.

The model calculates delay predictions based on the above-mentioned parameters and uses the GPS data to refine the predictions between established TIOS reference points. First, the reference time for position is calculated, using scheduled time of arrival and departure based on the pre-defined route plan, the train´s position progress and the timestamp. When GPS data is available, the model uses it to improve the delay predictions for the nearest OCP. The arrival time on the current segment is then calculated where the operational reserve is being considered. The use of GPS data ensures that real-time information about the train's location and speed is incorporated into the predictions, making them more accurate. Calculations for further propagation on the subsequently segments then follows. An example on further propagation calculations for the specific case can be found in figure 5.5.5. Additionally, the original figures prevented by Funkwerk, achieved from Bane NOR, can be found in the appendix C.



Example propagation calculation:

Segment	Prognosed departure:	Prognosed arrival:
C ⟶ D		10:38:20 (+ 05:20)
D ⟶ E	10:39:20 (+ 04:20)	10:50:50 (+ 02:50)
E ⟶ F	10:51:50 (+ 02:50)	11:02:35 (+ 01:35)
F ⟶ G	11:02:35 (+ 01:35)	11:08:50 (+ 00:50)

Figure 5.5.5: Example on propagation delay estimation based on the calculations from the specific case.

5.5.6 Performance Metrics

Two common performing metrics were used in both analysis and feature engineering phase, to determine whether the model was a good fit to the data or not. Further, the metrics gives a review of the models them self, in addition to being good indicators when comparing with other prediction models, both in this and in other researches.

The performance metrics mean average error (MAE) and root mean square error (RMSE) (JJ 2022) are described below and shown in the respectively equations 5.9 and 5.10. When taking into account the predicted and observed times, the prediction models' accuracy is assessed using both MAE and RMSE. These parameters, according to Handelman et al. (2019), are a gauge of how well the regression line fits the data and produces accurate predictions.

MAE, Mean Average Error, determines the average magnitude of the errors in the models' predictions without accounting for their directional component. When considering the average of the absolute differences between prediction and actual observation in the test sample, all individual deviations are given equal weight.

RMSE, Root Mean Square Error, is a quadratic scoring formula that determines average error magnitude. The average squared discrepancy between the predicted result and the actual observation is what determines this.

$$MAE == \frac{1}{N} \sum_{k=1}^{N} |\hat{y}_k - y_k| \tag{5.9}$$

$$RMSE == \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\hat{y}_k - y_k)^1} \tag{5.10}$$

where y_k and $\hat{y}_k$ respectively represents the actual delay and absolute estimate deviation. Both metrics can range from 0 to ∞ where a lower MAE and RMSE indicates more "accurate" predictions, lower "average loss" across data, and a better performance of the model. However, there isn't an universally accepted definition of a good score because MAE and RMSE are returned on the same scale as the predictions. The evaluation of the score is in other words defined within the context of the dataset. From the related research on train prediction models, MAE equals to When errors are squared before being averaged, RMSE lends comparatively significant weight to large errors. This shows that observations that are farther from the mean are more likely to cause the RMSE to increase. You can reach a low RMSE only by having both a high precision and no systematic error. Furthermore, the errors have a wider range and are more evenly distributed the further apart the MAE and RMSE are.

5.6 Methods and tools

The following methods and tools are used in data management, analysis and feature engineering phases.

Python is a popular computer language for data research and data management. Python was chosen for this project because of its large open-source modules that are appropriate for scientific computing and data analysis and its convenience of use due to its many uses, including ease of analysing and organizing the data. In this project, libraries like Pandas, NumPy, Matplotlib, and Pythouch are used.

Jupyter is a web-based notebook interface made by JupyterLab. It was a useful tool to use because of its programming approach, which allows for the execution of specific sections of code without running the entire program. The interface's ability to work with a variety of tools and programs is helpful in a number of circumstances, including the collecting, cleaning, analysing, and displaying of data.

Pandas is an open source Python library that supports processing tabular data and is based on the Python programming language. As it supports data processing, cleansing, and munging, Pandas is one of the fundamental libraries in any workflow for data science projects.

BeatifulSoup is a Python library to extract data from HTML and XML files. To enable idiomatic means of exploring, finding, and updating the parse tree, it collaborates with external parsers. The library and the ’lxml’ parser were used to navigate and search through the data throughout the data cleaning process, which greatly reduced the amount of time required.

Plotly is the Python library for interactive data visualizations, effective at describing and examining data. Trends, patterns, and linkages in the data collection are easier to identify, comprehend, and express thanks to the visualization tool. Box plots have been created using Plotly.

NumPy is a library in python which offers mathematical functions, random number generators, linear algebra routines, and more. NumPy is often used working with arrays, and the functionality within the domain of linear algebra makes it highly relevant for machine learning projects.

Google Maps API include a set of programming interfaces and tools provided by Google which allows developers to embed Google Maps in their own applications. It has been utilized for geo-coding, which means converting addresses (or names of train stations in this example) into a set of geographic coordinates.

PyTorch is an open source machine learning framework in python. It is used for building and training deep neural networks. PyTorch offers several key features including support for GPU acceleration, dynamic computation allowing network architecture modification on-the-fly, and it has a large community which provides access to vast amounts of resources.

Scikit-Learn is a popular open-source machine learning library in python. It is built on top of NumPy, SciPy and matplotlib, and provides a range of tools and algorithms for data analysis. The algorithms include regression, clustering and dimensionality reduction algorithms which are commonly used in the feature engineering phase of a machine learning project.

5.7 Pipeline

To provide a comprehensive overview of the project from start to finish, a visual representation tool known as a *pipeline* has been developed to demonstrate all the stages involved. This is visually depicted in figure 5.7.1. The green pipes in the centre of the illustration, with arrows flowing from left to right, represent the main path from the start to the end. These are the primary procedures. Different processes are outlined for each green pipe, one for each data set process. The figure is also divided to signify two main categories: *data retrieval* and *data management* to the left, and *analysis* to the right. This division aids in delineating the various stages of the project for a clearer understanding.

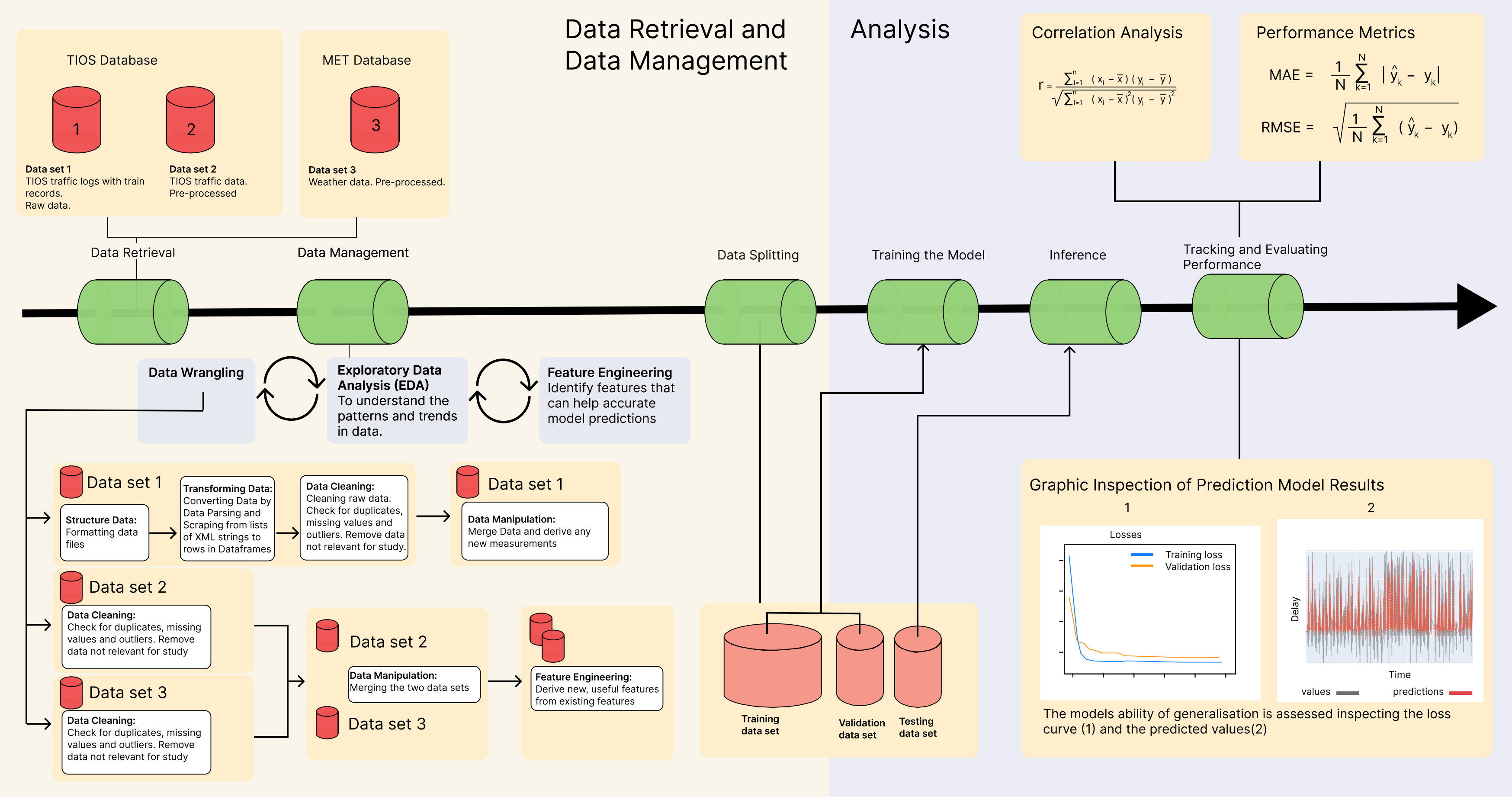

Data Retrieval and Data Management
TIOS Database
1
2
Data set 1
TIOS traffic logs with train records.
Raw data.
Data set 2
TIOS traffic data.
Pre-processed
MET Database
3
Data set 3
Weather data. Pre-processed.
Data Retrieval
Data Management
Data Splitting
Data Wrangling
Exploratory Data Analysis (EDA)
To understand the patterns and trends in data.
Feature Engineering
Identify features that can help accurate model predictions
Data set 1
Structure Data:
Formatting data files
Transforming Data:
Converting Data by Data Parsing and Scraping from lists of XML strings to rows in Dataframes
Data Cleaning:
Cleaning raw data. Check for duplicates, missing values and outliers. Remove data not relevant for study.
Data set 1
Data Manipulation:
Merge Data and derive any new measurements
Data set 2
Data Cleaning:
Check for duplicates, missing values and outliers. Remove data not relevant for study
Data set 3
Data Cleaning:
Check for duplicates, missing values and outliers. Remove data not relevant for study
Data set 2
Data Manipulation:
Merging the two data sets
Data set 3
Feature Engineering:
Derive new, useful features from existing features
Training data set
Validation data set
Testing data set
Analysis
Correlation Analysis
$r = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$
Performance Metrics
$\mathrm{MAE} = \frac{1}{N} \sum_{k=1}^{N} |\hat{y}_k - y_k|$
$\mathrm{RMSE} = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (\hat{y}_k - y_k)}$
Training the Model
Inference
Tracking and Evaluating Performance
Graphic Inspection of Prediction Model Results
1
2
Losses
Training loss
Validation loss
Delay
Time
values
predictions
The models ability of generalisation is assessed inspecting the loss curve (1) and the predicted values(2)

Figure 5.7.1: Python pipeline of the project from start to end.

CHAPTER

SIX

RESULTS

Results from each phase in the study are presented in this chapter. First, the results from the LSTM train delay prediction models and their performance statistics are presented. Then, the analysis of Bane NOR´s model is presented. Finally, some graphic inspection with comparison of the two results is presented. Outliers in the data set is removed using the boundary values presented in section 5.2.2.1. Likewise, delays exceeding the boundaries is removed from the data used in the analysis of Bane NOR's model.

6.1 LSTM Model

This section presents the results from the LSTM models which has been developed and trained on the *TIOS pre-processed data.* As mentioned in section 5.3, two separate models have been trained: the LSTM model, which does not include the delay classification feature, and the $\overline{LSTM}$ model, which includes the delay classification feature. Both these models have in turn been trained separately with weather data included in the feature space, and without it. This was done to evaluate the effect of including an external factor, weather, aiming to increase the predictive accuracy of the model. The results include learning curves from the training phase of the $\overline{LSTM}$ models, which are used to evaluate their ability to generalize on the data. Additionally, results from the correlation analysis will be presented. Finally, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) for both models will be calculated and presented. As the learning curves for the LSTM model and the $\overline{LSTM}$ model were nearly identical, results from the $\overline{LSTM}$ model is presented below.

6.1.1 With Weather Data

In this subsection, the results of the $\overline{LSTM}$ model that includes weather data in the feature space are presented. Figure 6.1.1 shows the learning curves for this model during the training phase. The figure demonstrates that as the number of epochs rises, the training and validation losses converge, demonstrating that the model is successfully learning from the data.

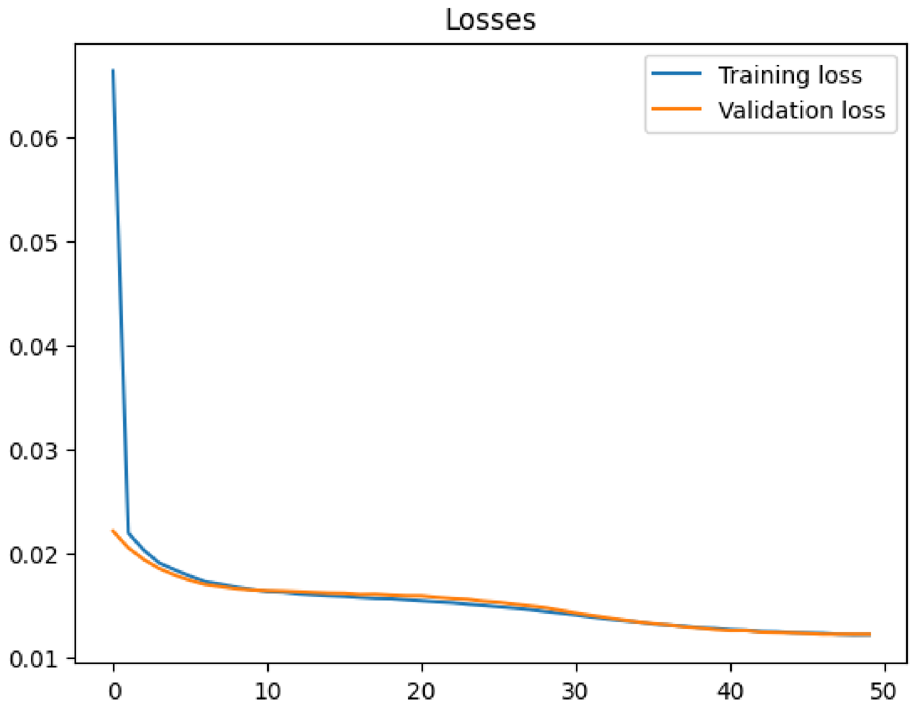


Figure 6.1.1: Learning curve for the $\overline{LSTM}$ model. Training data includes weather data.

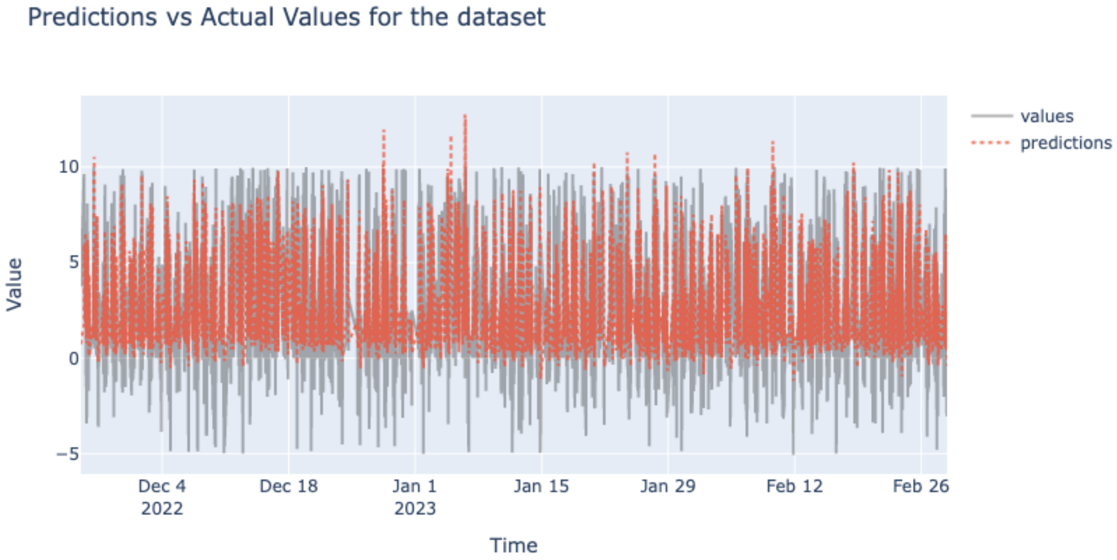


Figure 6.1.2: Comparing the predicted delays with the actual values.

Figure 6.1.2 presents a plot comparing the predicted train delays with the actual values. Visually analyzing the plot allows one to evaluate the general accuracy of the predictions and observe any patterns or disparities between expected and observed delays. Upon close examination of the plot, it becomes evident that the model has barely predicted any negative values. While 8 % of the actual values are negative, only 0.5 % of the predictions are negative.

Table 6.1.5 presents the MAE and RMSE values for the model with weather data. These metrics provide a quantitative measure of the model's performance in predicting train delays. In order to assess the effects of including weather condition information, it is essential to compare these values with the model that does not contain weather data, where a lower number for both MAE and RMSE implies better and more accurate performance.

6.1.2 Without Weather Data

In this subsection, the results of the $\overline{LSTM}$ model that does not include weather data in the feature space are presented. Figure 6.1.3 shows the learning curves for this model during the training phase. The training and validation losses converge as the number of epochs rises, similar to the model using weather data, indicating that the model is also successfully learning from the data.

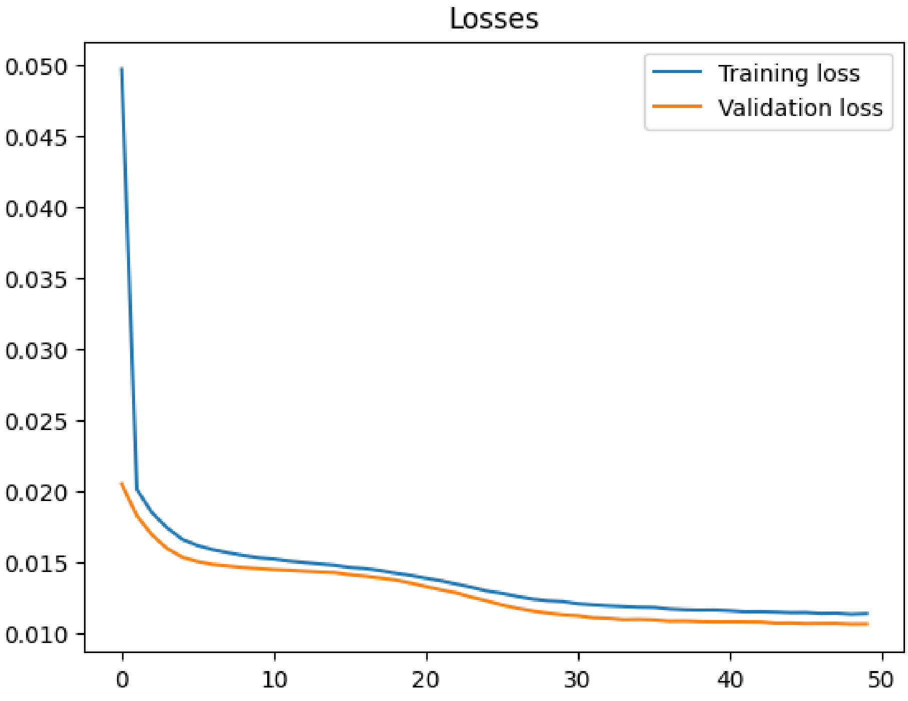


Figure 6.1.3: Learning curve for the $\overline{LSTM}$ model without weather features included in data set.

Figure 6.1.4 presents a plot comparing the predicted train delays with the actual values. Similar to 6.1.2, the model without weather data also tends to predict positive delays. Although the training data contain a small percentage of negative delays, the portion of negative predictions are far smaller than for the actual values. This finding will be discussed in the discussion.

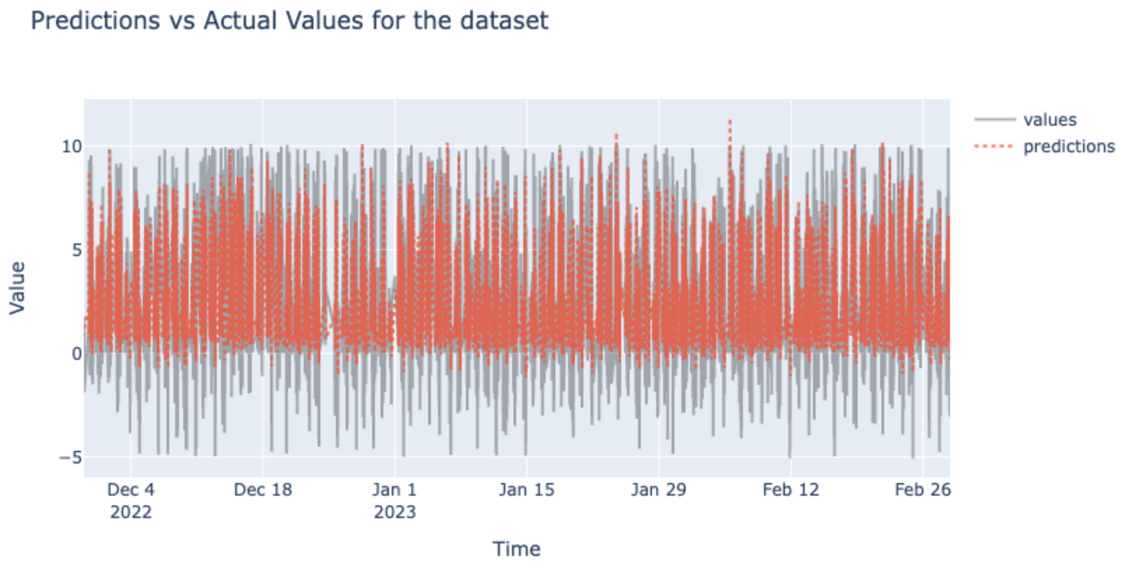


Figure 6.1.4: Comparing the predicted delays with the actual values.

6.1.3 Hyper-parameter Tuning

Table 6.1.1 shows the performance of the $\overline{LSTM}$ model for four different hyper-parameter configurations. These configurations represent the best-performing among numerous other configurations calculated using grid search. The table displays the selected hyper-parameters from table 5.5.3, including learning rate, number of LSTM layers, number of hidden units, number of epochs, dropout, and data-split, as well as the resulting mean absolute error (MAE) and root mean squared error (RMSE) for each configuration.

Hyper-parameter	Config 1	Config 2	Config 3	**Config 4**
Learning rate	0.001	0.0001	0.0001	**0.0001**
Number of LSTM layers	4	4	3	**3**
Number of hidden units	64	128	64	**64**
Number of epochs	30	40	50	**50**
Dropout	0.2	0.1	0.2	**0.2**
Data-split	70-15-15	70-15-15	70-15-15	**80-10-10**
MAE (min)	1.18	1.19	1.21	**1.18**
RMSE (min)	1.73	1.69	1.70	**1.70**

Table 6.1.1: Hyper-parameter configurations and performance

It is essential to note that the differences in performance between the four configurations are relatively small. However, the best-performing configuration, Configuration 4, is chosen for its slightly better performance in terms of both MAE and RMSE.

These results demonstrate the importance of hyper-parameter tuning for LSTM models, as it can help identify configurations that yield improved prediction accuracy. The grid search method used in this study allowed for an extensive exploration of the hyper-parameter space, and the inclusion of only the top four configurations in the table highlights the effectiveness of the method in selecting the best-performing configurations.

6.1.4 Correlation

In this section, the correlation between the LSTM and the $\overline{LSTM}$ model's predictions and the actual train delays are assessed, as well as the correlation between weather features and train delays. This analysis provides insights into the relationships between the variables and helps explain the performance differences between the two $\overline{LSTM}$ models: one with weather features and one without. Additionally, correlations are calculated between actual train delays, predictions and the lagged target value observations, before and after including the delay categorisation feature, delay_grad.

Firstly, Pearson correlation coefficients were computed between each weather feature, the actual train delays, and the LSTM model predictions.

	Actual Delay	Prediction
Temp	$r_{temp,d} = -0.071$	$r_{temp,p} = -0.137$
Prec	$r_{prec,d} = 0.014$	$r_{prec,p} = 0.005$
Snow	$r_{snow,d} = 0.016$	$r_{snow,p} = 0.022$
Wind	$r_{wind,d} = 0.056$	$r_{wind,p} = 0.107$

Table 6.1.2: Pearson correlation coefficients between actual delays, predictions, and weather features for the $\overline{LSTM}$ model.

All four weather features showed weak correlations with both actual delay values and predictions. Temperature (Temp) had a correlation of $r_{temp,d} = -0.071$ with actual delay and $r_{temp,p} = -0.137$ with predictions. Precipitation (Prec) showed a correlation of $r_{prec,d} = 0.014$ with actual delay and $r_{prec,p} = 0.005$ with predictions. Snow had a correlation of $r_{snow,d} = 0.016$ with actual delay and $r_{snow,p} = 0.022$ with predictions. Lastly, Wind had a correlation of $r_{wind,d} = 0.056$ with actual delay and $r_{wind,p} = 0.107$ with predictions.

The correlations between predictions, actual delay, and lagged delay observations were computed for both LSTM models, with and without weather data.

Subsequently, the delay classification feature were added to the correlation analysis. This addition was included to explore the potential influence of additional recent delay trends on the LSTM model's predictions and actual train delays.

When comparing the tables 6.1.3 and 6.1.4, the correlation coefficient between actual delays and predictions for the LSTM model with and without weather data has increased from $r_{pred_w} = 0.564$ and $r_{pred} = 0.618$ to $\bar{r}_{pred_w} = 0.694$ and $\bar{r}_{pred} = 0.735$,

	LSTM With	Weather	LSTM Without	Weather
	Actual Delay	Prediction	Actual Delay	Prediction
Actual Delay	$r_{del_w} = 1.000$	$r_{pred_w} = 0.564$	$r_{del} = 1.000$	$r_{pred} = 0.618$
Prediction	$r_{pred_w} = 0.564$	$r_{del_w} = 1.000$	$r_{pred} = 0.618$	$r_{del} = 1.000$
Time Lag 1	$r_{tl1_{wd}} = 0.547$	$r_{tl1_w} = 0.955$	$r_{tl1_d} = 0.604$	$r_{tl1} = 0.973$
Time Lag 2	$r_{tl2_{wd}} = 0.327$	$r_{tl2_w} = 0.620$	$r_{tl2_d} = 0.396$	$r_{tl2} = 0.649$
Time Lag 3	$r_{tl3_{wd}} = 0.210$	$r_{tl3_w} = 0.401$	$r_{tl3_d} = 0.210$	$r_{tl3} = 0.338$
Time Lag 4	$r_{tl4_{wd}} = 0.149$	$r_{tl4_w} = 0.263$	$r_{tl4_d} = 0.200$	$r_{tl4} = 0.295$
Time Lag 5	$r_{tl5_{wd}} = 0.113$	$r_{tl5_w} = 0.168$	$r_{tl5_d} = 0.182$	$r_{tl5} = 0.230$

Table 6.1.3: Correlation coefficients for the LSTM model with and without weather data

	$\overline{LSTM}$ With	Weather	$\overline{LSTM}$ Without	Weather
	Actual Delay	Prediction	Actual Delay	Prediction
Actual Delay	$\bar{r}_{del_w} = 1.000$	$\bar{r}_{pred_w} = 0.694$	$\bar{r}_{del} = 1.000$	$\bar{r}_{pred} = 0.735$
Prediction	$\bar{r}_{pred_w} = 0.694$	$\bar{r}_{del_w} = 1.000$	$\bar{r}_{pred} = 0.735$	$\bar{r}_{del} = 1.000$
Delay_grad	$\bar{r}_{dg_{wd}} = 0.273$	$\bar{r}_{dg_w} = 0.415$	$\bar{r}_{dg_d} = 0.288$	$\bar{r}_{dg} = 0.385$
Time Lag 1	$\bar{r}_{tl1_{wd}} = 0.547$	$\bar{r}_{tl1_w} = 0.836$	$\bar{r}_{tl1_d} = 0.604$	$\bar{r}_{tl1} = 0.855$
Time Lag 2	$\bar{r}_{tl2_{wd}} = 0.327$	$\bar{r}_{tl2_w} = 0.524$	$\bar{r}_{tl2_d} = 0.396$	$\bar{r}_{tl2} = 0.561$
Time Lag 3	$\bar{r}_{tl3_{wd}} = 0.210$	$\bar{r}_{tl3_w} = 0.338$	$\bar{r}_{tl3_d} = 0.271$	$\bar{r}_{tl3} = 0.399$
Time Lag 4	$\bar{r}_{tl4_{wd}} = 0.149$	$\bar{r}_{tl4_w} = 0.232$	$\bar{r}_{tl4_d} = 0.200$	$\bar{r}_{tl4} = 0.295$
Time Lag 5	$\bar{r}_{tl5_{wd}} = 0.113$	$\bar{r}_{tl5_w} = 0.137$	$\bar{r}_{tl5_d} = 0.182$	$\bar{r}_{tl5} = 0.230$

Table 6.1.4: Correlation coefficients for the $\overline{LSTM}$ model with and without weather data

respectively. Interestingly, for the model with weather data, the delay classification feature demonstrated a correlation of $\bar{r}_{dg_{wd}} = 0.273$ with actual delays and $\bar{r}_{dg_w} = 0.415$ with predictions, indicating a moderate positive relationship. However, the inclusion of the delay classification led to a slight reduction in correlation coefficients for the time lags. For instance, the correlation of Time Lag 1 with actual delays and predictions dropped to $\bar{r}_{tl1_{wd}} = 0.547$ and $\bar{r}_{tl1_w} = 0.836$, respectively. This reduction continued with each subsequent time lag, suggesting that the delay classification feature might be capturing some of the information that was previously accounted for by the time lags.

For the $\overline{LSTM}$ model without weather data, the delay classification feature demonstrated a correlation of $\bar{r}_{dg_d} = 0.288$ with actual delays and $\bar{r}_{dg} = 0.385$ with predictions. The correlation coefficients for the time lags also reduced upon the inclusion of delay classification, in a similar pattern observed in the model with weather data.

In summary, the inclusion of the delay classification feature appears to enhance the correlation between actual delays and predictions for both models. However, this addition also seems to reduce the correlation coefficients for the time lags, suggesting a shift in the models' reliance on these features. A comprehensive discussion and interpretation of these results, including the implications of the inclusion of delay classification, will be addressed in the Discussion chapter.

6.1.5 Performance Metrics

In this section, the performance of four LSTM models is evaluated. These models are variations of two base models, LSTM and $\overline{LSTM}$, each trained on data sets both with and without weather features. The model's performances are assessed using commonly employed performance metrics for regression tasks: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). These metrics help quantify the accuracy of the models in predicting train delays and facilitate a direct comparison of their performance.

The initial LSTM model, trained without weather features and excluding delay classification, achieved a lower MAE (1.41 min) compared to its counterpart trained with weather features (1.55 min). This suggests that, on average, the model without weather features produced smaller prediction errors. Similarly, the RMSE was lower for the model without weather features (1.98 min) versus the model with weather features (2.11 min), implying it was less prone to larger errors.

Upon the introduction of the delay classification feature, represented by the $\overline{LSTM}$ models, an improvement was observed in the performance of both weather-inclusive and -exclusive models. The $\overline{LSTM}$ model with weather features yielded an MAE of 1.32 min and an RMSE of 1.84 min, while the $\overline{LSTM}$ model without weather features exhibited superior performance, achieving an MAE of 1.18 min and an RMSE of 1.70 min.

Table 6.1.5 presents a summary of the performance metrics for all LSTM models:

Model	**MAE**	**RMSE**
LSTM with weather features	1.55 min	2.11 min
LSTM without weather features	1.41 min	1.98 min
$\overline{LSTM}$ with weather features	1.32 min	1.84 min
$\overline{LSTM}$ without weather features	1.18 min	1.70 min

Table 6.1.5: Summary of performance metrics for all four models.

In summary, the inclusion of the delay classification feature led to improved performance for both models. Even so, the models without weather features consistently outperformed the ones with weather features, in terms of both MAE and RMSE, regardless of the inclusion of the delay classification feature.

6.2 Bane NOR´s Model

Bane NOR´s train delay prediction model have been examined. Both the correlation coefficient and the precision performance have been measured.

Some initial work have been done in order to have models with performances that are as comparable as feasible. As described in section 5.4, estimates predicted more than fifty minutes and under five minutes before the departure have been removed from the data set. By excluding both early and late generated predictions, the two models are more comparable. Further, the same delay restrictions as the LSTM model have been put in place, excluding the bigger delays. The time limit constraint was put in place to replicate the estimation from the LSTM model, -5.075 and 10.005 minutes, respectively. In order to assess the performance precision of the model in general, it has also been assessed both without and with boundaries on an hour. Below follows an overview of the delay restrictions and the responding data set sample size.

The data set without delay boundaries, $[\infty, \infty]$, has a sample size equal to 1.187.845

The data set with an hour delay boundaries, $[-60min, 60min]$, has a sample size equal to 1.141.950

The data set with the same delay boundaries as the LSTM model, $[-5.075min, 10.005min]]$, has a sample size equal to 756.708

The results follows below.

6.2.1 Correlation

The Pearson correlation coefficient were calculated have been measured and analysed to examine the relationship between the actual delay, equation 5.1, and absolute estimate deviation, equation 5.3. The coefficients are used to evaluate the linear correlation for Bane NOR´s model on the train route between Oslo and Trondheim in the period between 15th of June 2022 and 22th of Mars 2023. A strong correlation indicates that the estimated departure time is close to the actual departure time.

Restrictions	**Pearson Correlation Coefficient**
$[\infty, \infty]$	$r_{[\infty,\infty]} = 0.745$
[-60 min, 60 min]	$r_{[-60,60]} = 0.892$
[-5.075 min, 10.005 min]	$r_{[-5.075,10.005]} = 0.660$

Table 6.2.1: Summary of the Pearson correlation coefficients for Bane NOR´s model with different restrictions

The Pearson correlation coefficient equation can be found in 5.6 followed by an evaluation guide. Based on this evaluation guide, the following can be said about the coefficients:

$r_{[\infty,\infty]}$ The Pearson correlation coefficient without boundaries is a strong correlation.

$r_{[-60,60]}$ The Pearson correlation coefficient with an hour boundaries is a very strong correlation.

$r_{[-5.075,10.005]}$ The Pearson correlation coefficient with the same boundaries as the LSTM model is a strong correlation.

The Pearson correlation coefficient was measured to be $r_{[\infty,\infty]} = 0.745$ when including all estimates and delays. This is a strong coefficient. When excluding delays above an hour before and after, the Pearson correlation coefficient was measured to be $r_{[-60,60]} = 0.892$. The coefficient is considered as very strong. Further, the delay boundaries were decreased to the same as the LSTM train delay prediction model, -5.075 and 10.005 minutes, respectively. The Pearson correlation coefficient was then measured to be $r_{[-5.075,10.005]} = 0.660$. One can observe that when the delays decreases, the correlation decreases from strong to moderate. All three Pearson correlation coefficients have a positive correlation between the two quantitative values, meaning that when a delay increase the estimated departure time also increases.

6.2.2 Performance Metrics

The overall prediction precision of the model have been examined through the performance metrics MAE and RMSE, equation 5.9 and 5.10, further described in section 5.5.6.

Restrictions	**MAE**	**RMSE**
$[\infty, \infty]$	4.99 min	17.85 min
[-60 min, 60 min]	3.46 min	6.311 min
[-5.075 min, 10.005 min]	1.79 min	2.55 min

Table 6.2.2: Summary of performance metrics for Bane NOR´s model with different restrictions

The performance metrics illustrates how the models accuracy varies depending on the delays. The accuracy tempt to increases when the delay restrictions decreases.

MAE describes the average error. An MAE between 1-4 minutes is not very good, but an acceptable average error. Both MAE´s with restrictions are therefor acceptable errors. The model is however not very accurate when accepting the bigger delays too.

Big RMSE indicates that there are large errors. Large errors are given a relatively high weight by RMSE, since they are squared before being averaged. This demonstrates that observations that deviate more from the mean are more sensitive to the RMSE. The big RMSE indicates that there are large errors. Additionally, errors have a wider range and are more evenly distributed the farther apart the MAE and RMSE are from one another. The big span between the metrics indicates that the errors are distributed on all of the data set. This applies in particular to the performance metrics without and with an hour boundaries.

6.3 Model Comparison

New plots containing the output from both models have been formed in order to make it easier to compare and analyze the two models. First, a summary of the correlation coefficients and performance indicators is gathered and given in two tables.

6.3.1 Model Performance Overview

The tables provide a clear picture of the relationships between model strength and accuracy. The Pearson correlation coefficients are displayed in Table 6.3.1. The results of the accuracy metrics are summarized in Table 6.3.2 gives an overview of the precision metrics outcomes.

Model	**Pearson Correlation Coefficient**
LSTM with weather features	$r_{pred_w} = 0.564$
LSTM without weather features	$r_{pred} = 0.618$
$\overline{LSTM}$ with weather features	$\bar{r}_{pred_w} = 0.694$
$\overline{LSTM}$ without weather features	$\bar{r}_{pred} = 0.735$
Bane NOR $[\infty, \infty]$	$r_{[\infty,\infty]} = 0.745$
Bane NOR $[-60min, 60min]$	$r_{[-60min,60min]} = 0.892$
Bane NOR $[-5.075min, 10.005min]$	$r_{[-5.075min,10.005min]} = 0.660$

Table 6.3.1: Overview of the Pearson correlation coefficient for both models.

Model	**MAE**	**RMSE**
LSTM with weather features	1.55 min	2.11 min
LSTM without weather features	1.41 min	1.98 min
$\overline{LSTM}$ with weather features	1.32 min	1.84 min
$\overline{LSTM}$ without weather features	1.18 min	1.70 min
Bane NOR $[\infty, \infty]$	4.99 min	17.85 min
Bane NOR [-60 min, 60 min]	3.46 min	6.311 min
Bane NOR [-5.075 min, 10.005 min]	1.79 min	2.55 min

Table 6.3.2: Overview of the performance metrics for both LSTM models, and Bane NOR´s.

6.3.2 Visual Inspection

Visual inspection is a qualitative assessment technique used to analyse the estimate deviation for the two models. Box plots are used as the visual tool to easier see the spread of the deviations, the normality and its outliers. Box plots are described in section 5.4.1.1. Further, the box plots were provided by *plotly*, see section 5.6. *Plotly* gives valuable insight of the descriptive numerical values, like medians, highest maximum and lowest minimums of each box.

When comparing the two data sets, only stations for passenger transit have been examined. The base stations, where trains don't stop, have been removed from TIOS raw data set.

Estimate deviation, equation 5.2, have been examined to visual inspect the performance of the two models. The deviation demonstrates how far the model's estimations are off based on the observed departures. The blue boxes show the estimate divergence from the LSTM model, while the red boxes show the estimate deviation from Bane NOR's model.

x-axis The horizontal axis represents each station from first to last, presented from *Oslo S* to *Trondheim.*

y-axis The vertical axis represents the delay, displayed in minutes.

The condition of the position of a box and its characteristics in the box plot indicates the following:

$$\forall points \in boxplot = \begin{cases} > 0 & \text{etd predicted to be earlier than atd} \\ = 0 & \text{etd predicted to be at same time as atd} \\ < 0 & \text{etd predicted to be later than atd} \end{cases}$$

When generating box plots with the Python library *Plotly*, described in section 5.6, the characteristics of a box is presented. For each box, the median have been collected to measure the average deviation for all stations. The average deviation formula follows:

$$AverageDeviation = \frac{1}{n}\sum_{j=1}^{n}|x_j| \tag{6.1}$$

where n is number of stations and $|x_j|$ is the absolute value of the median on each station.

Figure 6.3.1 examines the estimation deviation for each model without any limitations. This figure gives an overview of the models span of estimation deviation when there are no restriction other than the models restriction them self.

Figure 6.3.1 illustrates how much more distributed the estimate deviation from Bane NOR´s model is. When examining Bane NOR´s model, the absolute estimate deviation average median is 1.74 minutes, which is 104.4 seconds. The box plot figure shows how the model has some very big deviations on *Oslo S.* When examining the estimate deviation average median , but excluding Oslo S, the deviation is 1.28 minutes, which is 76.8 seconds. However, as the figure illustrates, Bane NOR´s model have more distributed and higher average deviation on the whole route than the LSTM prediction model. The estimate deviation average median for the LSTM model is 0.39 minutes, which is 23.42 seconds.

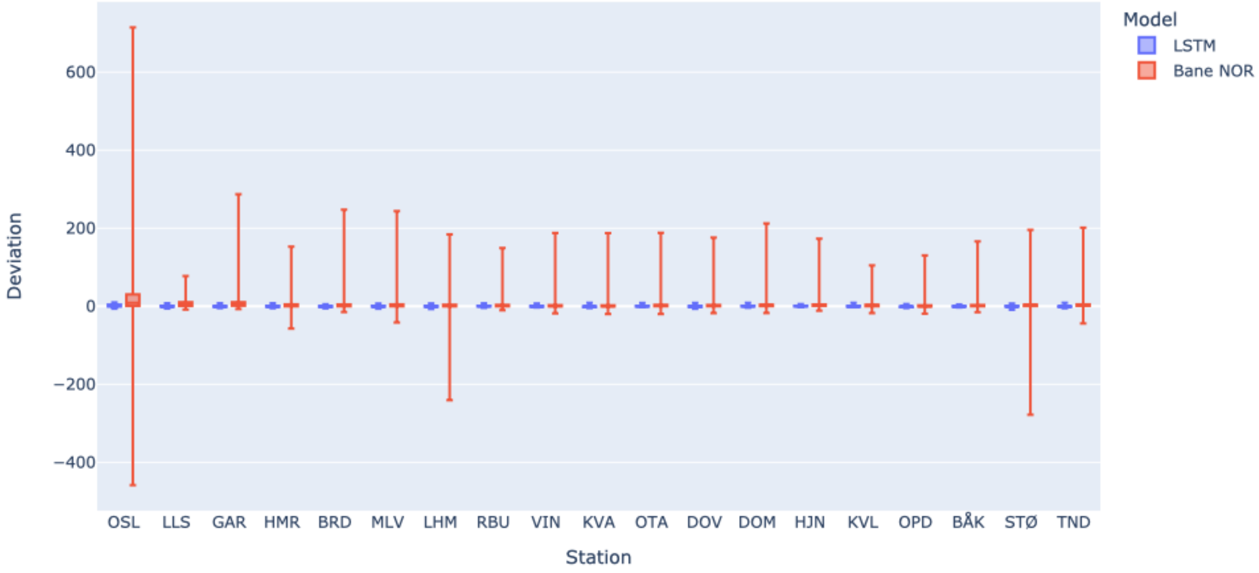


Figure 6.3.1: Box plot showing the distribution of the quantitative values of estimate deviation for both models, plotted for each station.

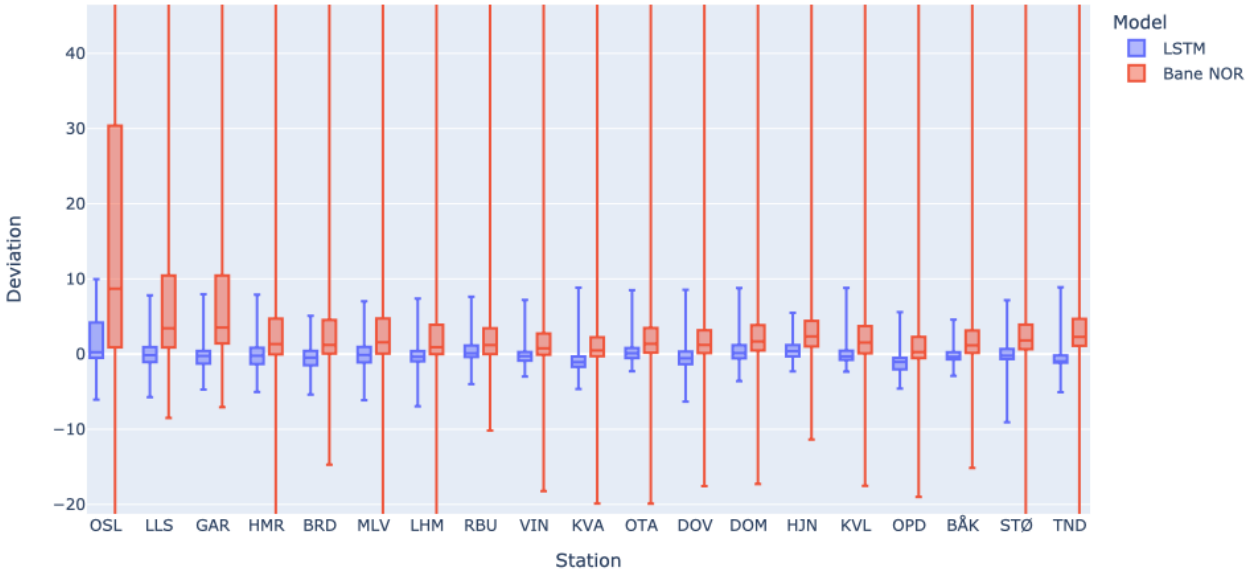


Figure 6.3.2: Box plot showing the distribution of the quantitative values of estimate deviation for both models, plotted for each station, zoomed in on the boxes.

Figure 6.3.2 gives a more reflective overview of the interquartile range of each box, as it is zoomed in. The figure also examines the estimation deviation for each model without any limitations.

Figure 6.3.3 also examines the estimation deviation for each model, but the deviations below 10 and above 15 minutes are removed. This has more an influence on the performance outcome of Bane NOR´s model. However, when removing the outliers, the characteristics of each box plot is easier to observe.

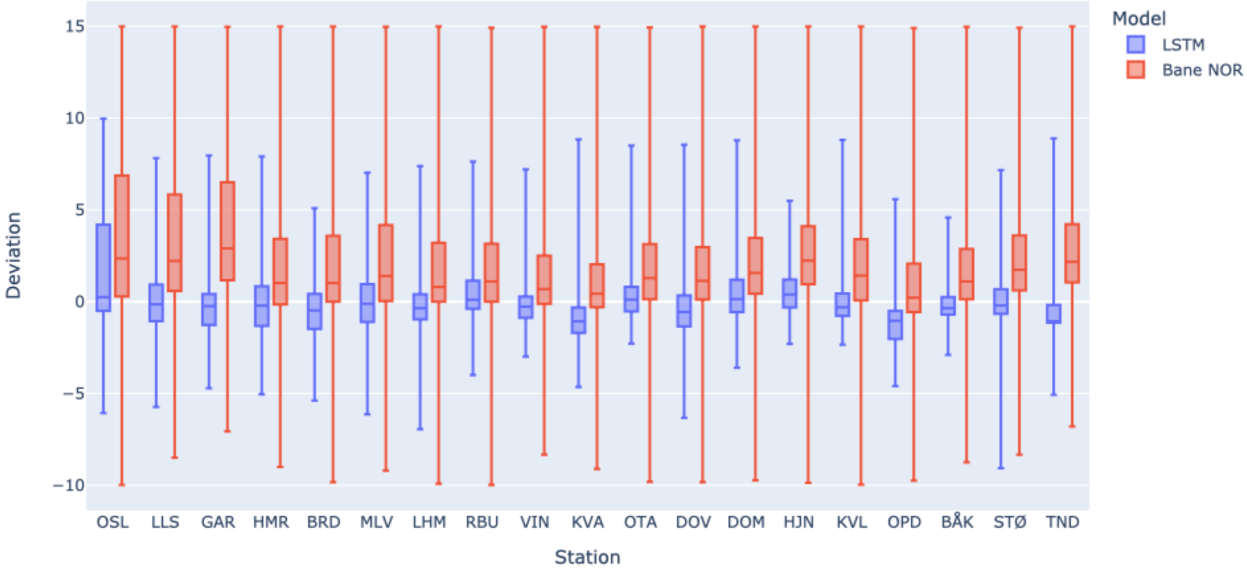


Figure 6.3.3: Box plot showing the distribution of the quantitative values of estimate deviation for both models, plotted for each station. Estimation deviation boundaries is here set to 10 minutes before and 15 minutes after.

The figure illustrates how only Bane NOR´s model is affected when removing estimate deviations below and above the boundaries. Neither maximum nor minimum estimate deviation by the LSTM model reach 10 in absolute value. The model´s highest maximum is 9.96546 minutes from the station *Oslo S*, and lowest minimum is -9.070591 minutes from the station *Støren*.

The estimate deviation average median with boundaries for Bane NOR´s model is measured to be 1.41 minutes, which is 84.5 seconds. The average deviation is the same as the average deviation without boundaries, but excluded Oslo S. Further, when excluding Oslo S measuring the estimate deviation average median with boundaries, it is measured to be 1.36 minutes, which is 81.4 seconds. The two figures and its measures indicates that Oslo S is an problematic station for the prediction model to Bane NOR. The estimate deviation average median with boundaries for LSTM prediction model is measured to be the same as without boundaries, namely 0.39 minutes.

Finally, figure 6.3.4 presents an overview of the interquartile range of each box with the estimate deviation boundaries. The figure illustrates how the Bane NOR´s model have a positive trend for their estimate deviation, while the LSTM model trend is more negative. This can also be seen when studying the medians.

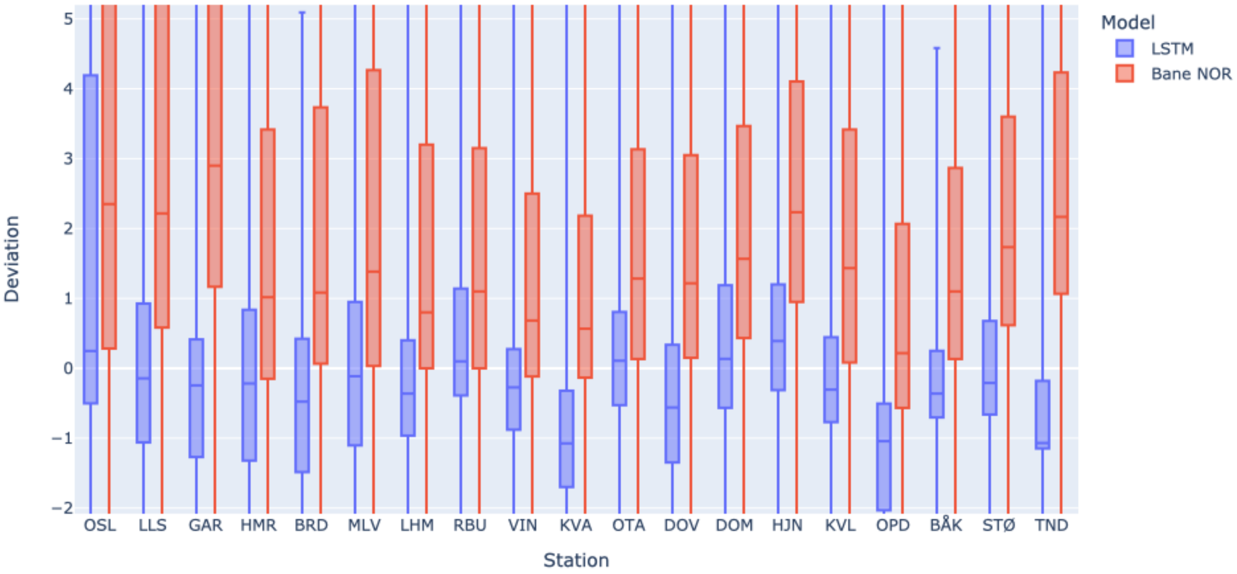


Figure 6.3.4: Box plot showing the distribution of the quantitative values of estimate deviation for both models, plotted for each station. Estimation deviation boundaries is here set to 10 minutes before and 15 minutes after.

CHAPTER

SEVEN

DISCUSSION

The results presented in the previous chapter Results, and the research questions, introduced in the chapter Introduction, are discussed below. It is advised to actively use the referred result tables and plots when reading the discussion. Additional supplementary and plots can be found in the appendix. In this way, the appendix substantiate the findings by providing a more comprehensive picture.

7.1 LSTM Prediction Model

This section discusses the findings of the research and provides an interpretation of the results in the context of predicting train delays. The main focus is on the comparison between the LSTM model with weather data on the one side, and the model without weather data on the other. Additionally, the influence of various pre-processing techniques and alterations on model architecture and hyper-parameters will be discussed in light of the obtained results.

7.1.1 Incorporation of Weather Features

Four weather features were included in the feature space of the data set for this project: temperature, precipitation, snow depth and wind. These were all included in the hope of improving the predictive accuracy of the LSTM model. The idea was to add a layer of context that are often important in transportation scenarios. Weather features can have a significant impact on train operations. By incorporating these weather features, the LSTM model could account for external factors of delay, making its predictions more robust and realistic. However, the results clearly indicate that the LSTM model's performance worsened when weather features were included in the feature space, in contrast to the theoretical expectations, provided in section 2.3. As indicated by the Pearson correlation coefficients in table 6.1.2, these weather features have an insignificant correlation with both the actual train delays and the predictions. Despite the strong theoretical basis suggesting that weather conditions could impact train delays, the model incorporating weather features performed worse than the one without these features. In fact, the MAE and RMSE decreased when weather features were removed, illustrated in the table 7.1.1 below.

Model	MAE	RMSE
LSTM with weather	1.32 min	1.84 min
LSTM without weather	1.18 min	1.70 min

Table 7.1.1: Performance metrics for the LSTM model with and without weather features.

This suggests that inclusion of the weather features seems to impair rather than improve the predictive performance of the model. Further, this prompts several possible interpretations and lines of enquiry.

This thesis seeks to precisely forecast small, everyday railway delays, as mentioned in RQ 1. The weather conditions taken into account in the model might not be the main causes of delays of this kind. This is consistent with the previous study by P. Wang and Q.-p. Zhang (2019) which illustrated how good weather not seem to have any notable impacts on the daily train delays. However, weather may play more of a significant role in causing substantial delays on congested networks or during extreme weather conditions. For instance, high temperatures might only cause delays if they cross a certain threshold leading to track expansion, or large changes in snow depth during small time-windows might cause large delays. Similarly, as noted by P. Wang and Q.-p. Zhang (2019), snowy weather could have a bigger impact in locations which experience snowy weather more rarely. In such cases, simply including raw weather features may not be sufficient. Alternatively, a more nuanced incorporation of weather features could be considered, e.g. through categorisation of weather conditions using specified thresholds rather than raw weather data. In fact, including the raw weather features in the model may add noise to the data, which could increase the error in the model's predictions, leading to higher MAE and RMSE values. More importantly, these cases of extreme weather conditions potentially lead to substantial delays, which is not the focus of this particular prediction model. The more routine delays might be influenced by other features.

The mismatch between theory and results could also be attributed to the quality and completeness of the weather data used. There was missing data for certain time periods and locations along the railway line. Although data imputation techniques were employed, this did not account for every occurrence of missing data. Such deficiencies in the data could introduce noise and reduce the model's predictive power, especially if the patterns of missing data is not random.

In reaction to weather conditions, railway companies frequently implement mitigation measures. This may also lessen the effect of weather-related railway delays, making it a less important feature in the forecast model. Due to the complicated, possibly non-linear, and conditional nature of this relationship, translating the theoretical knowledge of weather's influence on train delays into a prediction model has proven to be a challenging task.

In addition to an alternative incorporation of weather features, future research could also include consideration of larger delays, rather than purely focusing on small, everyday delays. Also, addressing the potential data quality issues, ensuring weather data is accurate and complete, will be crucial for improving the model's performance.

7.1.2 Feature Contribution on Model Performance

Several features were extracted from the original time series data set, delivered from Bane NOR, using various feature engineering techniques. It has been found that some features yielded more statistical significance than others. Firstly, the lagged observations, time-lags, of the target variable achieved high correlation coefficients. This aligns with the findings from the study by Sørskår (2022) which underlines the importance of past delay patterns when predicting future delay, as presented in section 3.6. Secondly, delay classification from the previous non-stop station, described in 5.3, also achieved moderate correlation coefficients. Both features contributed to higher predictive accuracy in terms of the performance metrics displayed in table 6.1.5. This demonstrates the predictive power embedded in past delay patterns, further emphasizing the importance of historical data in forecasting future outcomes. The time-lags provide delay patterns from the previous 5 stop station, which help capturing the delay propagation pattern along the railway line.

There are a varying number of base stations in-between certain stop stations, which are not part of these time-lags. As described in section 5.3, the data from these stations were removed from the training data as they introduce negative delays in the data. These kinds of delays are undesirable due to the inconvenient outcomes of trains leaving the stations earlier than scheduled. However, information about the time status of the train at these stations serve as a valuable pointer to a probability of departing on time at the next stop-station. This feature was included at a later stage of the modeling process, to illustrate the effect of the delay classification between the stations.

Model	Type	with weather	without weather
$LSTM$	Actual Value	$r_{pred_w} = 0.564$	$r_{pred} = 0.618$
	Time Lag 1	$r_{tl1_w} = 0.955$	$r_{tl1} = 0.973$
$\overline{LSTM}$	Actual Value	$\bar{r}_{pred_w} = 0.694$	$\bar{r}_{pred} = 0.735$
	Delay_grad	$\bar{r}_{dg_w} = 0.415$	$\bar{r}_{dg} = 0.385$
	Time Lag 1	$\bar{r}_{tl1_w} = 0.836$	$\bar{r}_{tl1} = 0.855$

Table 7.1.2: Overview of the discussed Pearson coefficients between the predictions for the LSTM and $\overline{LSTM}$ models, and their features.

Two distinct sets of tables show the correlation coefficients of the feature space after the models had been trained, both with and without weather data. From table 6.1.3 and 6.1.4 one can easily observe that some of the statistical relationship between the target variable and the time-lags are shifted to this delay classification feature, delay_grad. The discussed coefficients are summarized in table 7.1.2.

The Pearson coefficient between time lag 1 and the predictions for the first modelling iteration, $LSTM$, shows a very strong correlation $r_{tl1_w} = 0.955$ and $r_{tl1} = 0.973$, respectively. When including the delay classification in the feature space, $\overline{LSTM}$, the coefficient decreased to $\bar{r}_{tl1_w} = 0.836$ and $\bar{r}_{tl1} = 0.855$. In spite of the decrease, the correlation coefficients are still considered strong. The importance of the time lag features for the LSTM model without delay classification is proved by this change in correlation. After the delay classification is implemented in the $\overline{LSTM}$ model, the relationship between the features are more distributed. Further, the change also illustrates how the delay classification feature is capturing meaningful information about the train delays, as expected. However, the varying Pearson coefficients underlines the complex inter-dependencies between features in this predictive task. When comparing the two LSTM models, the relationship between the prediction values and the actual values are more similar for $\overline{LSTM}$ than $LSTM$. This indicates that the model with delay classification is closer to the real world historical data when evaluating correlation.

The overall Pearson correlation coefficient between the actual and predicted value for the two prediction models also differ. The $\overline{LSTM}$ have a stronger correlation than LSTM, which can be seen when comparing $\bar{r}_{pred_w} = 0.694$ and $\bar{r}_{pred} = 0.735$, with $r_{pred_w} = 0.564$ and $r_{pred} = 0.618$. Further, the $\overline{LSTM}$ without weather feature have the strongest correlation, indicating a more accurate prediction model.

In addition to the correlation analysis, the accuracy of the models were analysed through performance metrics, presented in table 6.1.5. The performance metrics also gives a reflective review of the models after inclusion of the delay classification in the feature space. $\overline{LSTM}$ improved both MAE and RMSE, both with and without weather feature. The MAE was improved with 0.23 minutes from the LSTM to $\overline{LSTM}$ model, both with and without the weather feature. The $\overline{LSTM}$ model without weather feature have an mean average error of 1.18 min, which is a insignificant one compared to other studies. The RMSE improved with 0.27 minutes for the models with weather features and 0.28 minutes for the models without. However, neither of the performance metrics for all four model editions are very high, illustrating that the errors are not too large.

There are a variety of factors contributing to train delays. While the aforementioned features contribute to capturing the delay patterns along the railway line, which mostly arise from delay propagation, there are other features which could potentially help capture more general, high-level patterns in the data. This could be broader, seasonal, and cyclic trends in the data. These kinds of features are presented in table 5.3.2, and include *Day_ of_ week*, *Week_ of_ year*, *Hour_ sine* and *Hour_ cosine*, and *Is_ holiday*. For instance, weekday travel patterns, particularly during rush hours, might differ from those on weekends. Additionally, certain times of the day might see more delays due to increased passenger congestion. Also, the week of the year might help the model capture broader seasonal trends. Holidays might also have an impact on travel patterns, often leading to either a increase or decrease in travel demand, depending on the nature of the holiday. However, the trains included in the data set are all long-distance trains. The impact of these features might therefore differ from short-travel scenarios. While weekday patterns still might influence delays, the impact might not be as pronounced as in short-travel scenarios. On the other hand, holidays and vacation periods could have a more significant impact on long-distance trains, as more people opt for long-distance travel during these times. Given the long duration of the travel, the impact of the hour of the day might be diluted as the journey spans across multiple hours. Yet, the departure times could still influence the likelihood of delays due to factors like peak-hour congestion.

The cyclic, seasonality-related features could provide valuable contextual information that helps the model capture more general, high-level patterns in the data. However, these features achieved insignificant correlation coefficients, all falling within the range $[-0.08, 0.04]$. A low correlation coefficient does however not necessarily imply that a feature should be discarded from the feature space. The correlation coefficient measures the linear relationship between two variables, but many real-world relationships are non-linear. Including such features could, for instance, help the model generalize better on unseen data. This could be due to a more comprehensive picture being captured, and could particularly be important in time series prediction tasks where conditions typically vary across different times and locations.

Removing outliers can help the model focus on the more common patterns in the data, which may lead to a better understanding of the general trends and relationships between the features. Also, as stated in 5.2.2.1, it was essential to remove outliers from the data before training the model in terms of prediction performance, as the purpose of the model was to predict certain types of delays. In this sense, the outliers introduce noise and removing them can potentially help the model fit the data more accurately. However, although not evident during the visual inspection of the data during the EDA, some of the delays that were considered outliers could have been related to specific higher-level trends. While the most substantial delays are most likely related to rare, unforeseen events, some could potentially take part in a broader trend in the data.

As highlighted in section 6.1, the LSTM model demonstrates a significant under-prediction of negative delays. Although negative delays comprise 8% of both the overall data set and the test set used for evaluation, a mere 0.5% of the model's predictions were negative. This under-representation can be attributed to the original data set's skewed distribution of negative and positive delays, which stems from operational constraints: negative delays, indicating early departures, can inconvenience passengers and are thus usually avoided. Consequently, the model has a tendency to overfit on the positive delay values, virtually neglecting the negative ones.

While there are techniques available to address such imbalances in regression tasks, these have not been applied in this case. Given the undesirability of negative delays, the model's bias towards predicting positive delays could be interpreted as a positive outcome in this specific context.

The results from the correlation analysis and the performance metrics calculated for the LSTM model will be further discussed when comparing the LSTM model with Bane NOR's model in section 7.3.

7.1.3 Model Architecture and Hyper-parameters

The results presented in table 6.1.1 underscore the importance of meticulous hyper-parameter tuning in machine learning applications. The table presents the performances of four distinct configurations of the LSTM model, with Config 4 demonstrating marginally superior performance. Although the differences may seem minimal, even minor improvements in MAE and RMSE can have significant implications in real-world scenarios. These small improvements may affect hundreds or thousands of commuters daily.

Configuration 4 defined a learning rate of 0.0001, three LSTM layers, 64 hidden units, or *neurons*, 50 training epochs, a dropout probability of 0.2, and a data split allocating 80%, 10% and 10% of the data set to the training-, validation-, and test set respectively. This configuration yielded the lowest MAE, 1.18 min and RMSE, 1.70 min. This highlights that the best results were achieved when using a lower learning rate, fewer LSTM layers, and a higher percentage of data for training. The optimal configuration suggests that a more careful and gradual learning approach, coupled with a simpler model architecture, is more suited for this specific prediction problem.

The four configurations presented in table 6.1.1 included the best-performing out of a total of 432 configurations generated using grid search. The grid search method enabled an extensive exploration of the hyper-parameter space, and the results provide a demonstration of the effectiveness of this method. However, there exist more hyper-parameters, and larger value ranges which could be explored. This generates a vast number of possible configurations which could provide further improvements, but would require more time and computational resources, and was deemed unnecessary for this project.

The significance of pre-processing techniques and architectural adjustments to the LSTM model can not be overstated. Prediction models require good predictors of the target value, and thus the feature engineering techniques employed in this thesis has been paramount for the model performance. Data normalization ensured that the model was not unduly influenced by features with larger scales. Also, the careful handling of extreme values, or *outliers*, has played a critical role in preserving the integrity of the data set for this particular model aiming to predict small, everyday delays. Moreover, the choice of LSTM as the model architecture plays into the time-series nature of train delay data. This is due to the LSTM's inherent ability to handle long-term dependencies and perform complex predictions.

7.2 Bane NOR´s Model

Bane NOR's existing train delay prediction model is analysed and discussed based on the result, both model performance and correlation, presented in section 6.2. Only estimates created in the time interval between 50 and 5 minutes prior to a station are analyzed.

7.2.1 The Pearson Correlation

The Pearson correlation coefficients between actual delays and absolute estimate deviation can be found in table 6.3.1. All three measured Pearson coefficients are positive, indicating that when the actual delay increases, the estimated delay also tends to increase. This is a crucial component of the prediction model, but also a type of prerequisite.

The train delay prediction model has a strong correlation when including all delays. The Pearson correlation coefficient on $r_{[\infty,\infty]} = 0.745$ indicates that the train delay prediction model estimates correlates to the actual train registrations. Further, the correlation increase to very strong when removing the biggest delays. With delay boundaries on 60 minutes, the Pearson correlation coefficient is measured to be $r_{[-60,60]} = 0.892$. This indicates a very strong linear relationship between the two variables. It is tempting to draw the conclusion that the train delay prediction model has a more linear connection with smaller delays based on the two outcomes. However, when measuring the correlation coefficient with the same outlier boundaries as the LSTM train delay prediction model, the linear relationship decreases. The Pearson correlation coefficient with a lower delay boundary on -5.075 minutes and an upper delay boundary on 10.005 minutes was measured to be $r_{[-5.075,10.005]} = 0.660$. The correlation coefficient, however, is considered as a strong correlation even though it is weaker than the two others.

Correlation does not, however, imply causation, which means that changes in one variable do not always result in changes in the other variable. The model will generate and update new estimates each time it will receive new train position progress information. Since the model is an event-driven model, it predicts the delay evolution, resulting in estimates for the following stations. When a train runs on schedule, the predictions will follow, indicating that it will stay on time. Upon slight deviations, delays will occur, and the model will update the estimates thereafter. Furthermore, a delayed train can manage to drive in time, getting back into its schedule. Early predicted estimates might therefore be outdated and will not match the actual arrival and departure times. In these cases, the actual delays and absolute estimate deviation will not imply causation. This will weaken the correlation where the linear relationship decreases, especially on the last stations of the route.

It is worth noticing that the sample size do matter when checking the correlation between two variables in a data set. As presented in section 6.2, the three representative data sets corresponding to each delay boundary has different sample size. Each sample size is, however, assumed to contain a representative number of data. The constraints also affect the information stored in the three samples. However, either the size, content, or both, seems to influence the correlation strength, which can be observed by the three measured Pearson´s coefficients. There can be several reasons to this variability. First, when including all delays, the data lack homogeneity of variance. Since the model generates estimates for several following stations, some estimates will be outdated and under- or overestimated. Additionally, the bigger delays will also have a wider range of wrong estimates, further from the linear regression line. Bigger delays will open up the possibility of greater variation in estimates, higher variability. However, $r_{[\infty,\infty]}$ shows a strong correlation, even though the bigger delays are included. That might be because early bigger delays might be more actionable and easier to predict for following stations. When comparing $r_{[\infty,\infty]}$ with $r_{[-5.075,10.005]}$, the linear relationship is weaker, even though it has lower variability. This might be because the model predicts estimates in minutes. Small variance have a bigger effect on smaller estimates, since it constitute a larger percentage. That could be a significant factor as the correlation coefficient decreases as the delay boundaries are further constrained. Additional influencing factors could be the difference in dynamics inherent in short delays compared to moderate or longer delays. This include operational recovery, where trains often make up for short delays by increasing speed slightly if conditions allow, without affecting the overall schedule significantly. Additionally, rail operators add slack time, which absorb minor delays and prevent them from cascading down the line.

It is also important to note that the samples stemming from the various delay boundaries which have been set are subsets of each other. Bane NOR data set with delay boundaries $[-5.075min, 10.005min]$ is a subset of Bane NOR $[-60min, 60min]$ and Bane NOR $[\infty, \infty]$, and Bane NOR $[-60min, 60min]$ is a subset of Bane NOR $[\infty, \infty]$. Considering the sample size reduction when introducing the delay boundaries, it is apparent that the sample from Bane NOR $[-60min, 60min]$ constitute a large percentage of the sample from Bane NOR $[\infty, \infty]$, as the sample size is only reduced from 1.187.845 to 1.141.950. This could explain the increase from $r_{[\infty,\infty]}$ to $r_{[-60min,60min]}$, as the samples that has been removed include unusually large delays not partaking in a general pattern in the data. Furthermore, the sample size of Bane NOR $[-5.075min, 10.005min]$, when introducing the delay boundaries also used in the LSTM model, decrease from 1.141.950 to 756.708, indicating that a larger portion of the data is removed. Upon removal, the correlation coefficient decrease to $r_{[-5.075,10.005]}$. This might indicate that narrowing the delay boundaries to only account for smaller, everyday delays decrease the sample size significantly, which further reduces the model's ability to capture the full variability of the delays. The scope of delays being analysed is reduced, but the noise is not removed. Consequently, the noise's impact on the correlation is bigger.

7.2.2 Precision metrics

The precision metrics MAE and RMSE, presented in table 6.2.2, gives an illustration of the accuracy of the prediction model. The three different delay boundaries have been examined for both metrics. The performance metrics shows a clear pattern, where the model´s accuracy increases when the boundaries decreases.

When the train delay prediction model is tested without delay boundaries, both performance metrics scores above the acceptable accuracy restriction. Here $MAE_{[\infty,\infty]}$ = 4.99 minutes and $RMSE_{[\infty,\infty]}$ = 17.85 minutes. This indicates that the model have both high average and many large prediction errors when including the longer delays. Additionally, the big span between the two metrics illustrates that the errors have a wider range and are more distributed. Large errors are particularly undesirable since correct estimations are essential for proactive and anticipatory regulation of the current railway traffic control and offer passenger information. Additionally, decreasing the delay boundaries reduces the gap between MAE and RMSE. As bigger delays are eliminated when the boundaries are narrowed, this is an expected outcome. The model predicts smaller estimates when the delays are less, and the errors would have a less skewed distribution.

When removing delays over an hour, both before and after, the precision metrics decreases to $MAE_{[60,60]}$ = 3.46 min and $RMSE_{[60,60]}$ = 6.31 min. These boundary constraints undeniably increase the precision of the model. However, $RMSE_{[60,60]}$= 6.31 min indicates some relatively large prediction errors. $MAE_{[60,60]}$ is less than 4 minutes, which is in the higher tier, but within an acceptable average error.

The precision metrics turns out to be $MAE_{[-5.075,10.005]}$ = 1.79 min and $RMSE_{[-5.075,10.005]}$ = 2.55 min, when analysing the model with the same delay boundaries used for the LSTM models. The prediction model's accuracy falls within acceptable bounds. The average error is below two minutes, and the standard deviation of residuals are two and a half minute, meaning the data is more concentrated around the line of best fit.

When compared to other related studies' prediction errors, found in table 3.5.1, the model's precision metrics without boundaries $MAE_{[\infty,\infty]}$ and $RMSE_{[\infty,\infty]}$, are remarkably higher. It is worth noticing that the train delay prediction Bane NOR uses today do not have any boundaries. Additionally, the model generates estimates throughout the whole route. This indicates that early stations have less estimates than the latest station on a train route. Further, the estimates will always take the current situation into consideration. This can explain why delay boundaries clearly reduce the larger errors, which can be seen in the drop between the precision metric values from $RMSE_{[\infty,\infty]}$ = 17.85 min to $RMSE_{[60,60]}$ = 6.31 min. When excluding the bigger delays, the gap between the scheduled and actual departure time decreases. The largest possible error can either be when the prediction model estimates that the train is on time, but it is delayed. The second option is that the train is on time, but the prediction model estimates a delay. The second option is less realistic when considering the constraints of the model. By introducing delay boundaries, you also exclude the larger errors. Further, when analysing the model without any boundaries, it is reasonable to infer that the larger errors, illustrated by $RMSE_{[\infty,\infty]}$, is most likely related to the bigger delays and the early estimates when a delay has not occurred yet. The model predicts that the train is running on schedule, but is delayed later on the route.

7.3 Model Comparison

This section will undertake a detailed comparison between the LSTM and Bane NOR's model. The discussion will initially focus on their divergent underlying modelling paradigms, event-driven and data-driven, exploring the implications of these differences on the utility and adaptability of the respective models. Then, their performance is discussed and compared to each other. The linear relationship, the model accuracy and the estimate deviation are all discussed.

7.3.1 Prediction Methodology

The two models are both short-term delay prediction models for the operational planning phase. However, they differ by their underlying modelling paradigm, being data-driven or event-driven, and the specific mathematical model used. Bane NOR's model is event-driven with a linear equation system, while the LSTM model is a data-driven function approximator. Both data-driven and event-driven approaches are assumed to be suitable for real-time train delay prediction (K. Y. Tiong et al. 2023a). However, the fundamental differences have some direct effects on the outcomes from both models, making comparability complicated. Both models have their advantages and disadvantages, further presented and explored below.

Bane NOR's event-driven model focuses on creating a dependency network of train events within the prediction horizon using infrastructure and operational data. This approach leverages a variety of data sources to calibrate model parameters, incorporating real-time information to generate estimates. The event-driven nature of the model allows it to capture the dynamic relationships between train events, making it possible to generate probabilistic predictions of delay evolution across multiple stations. Bane NOR's model generates a series of estimates for multiple following stations along the route, estimating the evolution of delays. These estimates can serve as a valuable decision-support tool for train dispatchers, enabling them to track delays across the rail network. Moreover, passengers benefit from having information about potential delays at multiple upcoming stations, enhancing their ability to plan their journeys. However, this model's estimates are not without limitations; as the prediction horizon extends in terms of both time and distance, its accuracy may decrease, making it challenging to keep up with dynamic changes occurring during a train ride. As presented in 3.5, a study by Corman and Kecman (2018) explored how an extended prediction horizon impacts the prediction accuracy. They found that the mean absolute error (MAE) of their prediction model increased when increasing the prediction horizon. Although the majority of predictions continued to be accurate, they found that huge variations were harder to foresee. This aligns with the challenge of keeping up with dynamic changes occurring during a train ride, and is reflected in the inaccuracies in the estimates generated for multiple stations ahead by Bane NOR's model.

On the other hand, the data-driven LSTM model is built and trained on historical train data, learning patterns and trends from this data to generate predictions on unseen data. This approach primarily relies on direct predictors, or explanatory variables, which are indicative of real delays. While the data-driven model is adept at identifying general patterns and capturing the complexities of historical train operation data, it generates deterministic single-value predictions for only the next station. These single-value predictions are straightforward to interpret, offering a clear perspective on the future development of train delays. The practical utility of single-value predictions is somewhat limited, as they only predict for the next station and thus may not meet the full information needs of passengers and train dispatchers. Additionally, the LSTM model incorporates the most recent train operation status data when generating predictions. However, these predictions overlook the sporadic appearance of disturbances within the prediction horizon and do not provide an indication of their uncertainty.

Both data-driven and event-driven approaches have their merits when it comes to real-time train delay prediction. While the event-driven model benefits from its ability to generate estimates for multiple stations, capturing the dynamic nature of train events, the data-driven model excels at processing multi-dimensional noisy data and identifying hidden patterns in the data. However, the divergent nature of these paradigms also leads to differences in the utility and adaptability of the respective models.

Bane NOR´s model uses real-time data to generate multiple-value estimates. By continuously updating the estimates as new information is fed into the model, it is able to maintain a high level of responsiveness to evolving circumstances. This real-time data utilisation enables the model to account for sudden changes, disturbances, or unforeseen events, making it more suitable for dynamic, real-world railway operations.

On the other hand, the LSTM model learns patterns and trends from historical train data which are then used to generate predictions for unseen data. It uses this historical data to approximate a function that maps a set of predictor variables to a target variable. Although the LSTM model is effective at detecting general patterns in the historical data, it may struggle to account for unique, day-to-day fluctuations or rapidly changing conditions that are not reflected in the training data.

One way to enhance the performance of the LSTM model is to integrate real-time data into the training process. As stated in section 3.2.2, an approach to adapting a pre-trained model to new data is called fine-tuning. By fine-tuning the LSTM model using real-time data, the model can capture more accurate and timely information about the current status of the train operations. This can provide a more precise representation of the current operational environment, potentially leading to more accurate delay predictions. As stated in section 3.2.2, Shon et al. (2022) proposed *continual learning* as an approach for fine-tuning a pre-trained model. Although the authors study this method with regards to image classification, a similar approach might be a viable solution towards continual fine-tuning of an LSTM, progressively updating the model parameters based on new incoming data. Real-time data integration allows the model to account for any sudden changes, disturbances, or unforeseen events which may not be captured in the original historical data used for training. Thereby it could serve to enhance the LSTM model's predictive ability and accuracy.

Moreover, every day of operations in a modern and complex railway system is unique. Although there exist some relationships and patterns which could be generalized by a machine learning model trained on historical data, there also exist unique patterns on a day-to-day basis. As presented in section 2.1.7, Gestrelius et al. (2015) identified 314 different patterns when studying historical data from a whole year of 365 days, further emphasising the variability of patterns emerging in railway systems. This also underlines the potential importance of training on real-time data when performing accurate real-time predictions. This does, however, depend on maintaining the correct balance between retaining past knowledge learned from the historical data, and adapting to new patterns provided by the real-time data. Continuous fine-tuning could improve the adaptability of the model, making it a more useful tool for real-time train delay prediction.

The LSTM model could also be extended to predict delays at multiple stations by incorporating real-time data and using a sequence generation approach. This method is akin to next-word prediction in language models, where the prediction of the next word depends on the previous words. Similarly, in the context of train delay prediction, the LSTM model can predict the delay for the next station based on the sequence of delays at the previous stations.

The input sequence length of such a model would have to be constant (N). So, at the first couple of stations where there are none or few previous delays to base its predictions on, the input sequence of delays could be padded with zeros, essentially letting it know that there have not been a full set of stations prior to the following station. For each station it predicts, this prediction would be added into the input sequence, and the earliest delay in the sequence is removed to keep the length constant. This process is repeated at each subsequent station on the route each time using the updated sequence of delays as input to the LSTM model. Alternatively, instead of padding the input sequence, this model could be used only from the Nth station and onwards after receiving enough real-time data to fill up its input sequence.

This approach essentially transforms the LSTM model into a generative model, generating a sequence of predicted delays for multiple future stations, one station at a time. As the train progresses along its route, and actual delay data becomes available, this data is used to update the input sequence. This ensures that the predictions for future stations are always based on the most recent data and helps to maintain the accuracy of the predictions.

7.3.2 Model Performance

Precision metrics and correlation analysis have been used to assess the models' performance. Box plots have also been generated to visually inspect the range of the estimate deviation for each station for both models. The measurements with the comparable delay bounds, [-5.075,10.005], serve as the starting point for comparison between the two models. An overview of all results discussed in this section can be found in section 6.3.

As stated in section 7.3.1, Bane NOR's model and the LSTM models differ in their prediction horizons. While Bane NOR's model generate estimates for multiple stations ahead, the LSTM model only predict the delay at the next station. This essentially means that Bane NOR's model has an $N : 1$ relation between the estimates and the actual delay, and the LSTM model has a $1 : 1$ relationship between the predicted and actual delay. This difference might have significant implications on model performance and interpretation of results.

7.3.2.1 Comparison of the Pearson Correlation Coefficients

All measured Pearson coefficients for both models are positive, and moderate or strong. Bane NOR´s model has higher Pearson correlation coefficient when compared with the LSTM model without delay classification. When implementing delay classification, the $\overline{LSTM}$ model have higher linear relationship.

LSTM with weather features, $r_{pred_w} = 0.564$

LSTM without weather features, $r_{pred} = 0.618$

$\overline{\textbf{LSTM}}$ with weather features, $\bar{r}_{pred_w} = 0.694$

$\overline{\textbf{LSTM}}$ without weather features, $\bar{r}_{pred} = 0.735$

Bane NOR with delay boundaries, $r_{[-5.075min,10.005min]} = 0.660$

The Pearson r for the LSTM model with weather features is a moderate correlation. The two other are considered as strong, where Bane NOR´s is $r_{[-5.075min,10.005min]} = 0.660$ and the LSTM without weather is $r = 0.618$. However, $r_{[-5.075min,10.005min]}$ linear relationship is not significant stronger than the two others.

Further, when including the delay classification in the $\overline{LSTM}$ model, the $\overline{LSTM}$ model has a stronger correlation than Bane NOR´s. The Pearson coefficient for the $\overline{LSTM}$ model with weather feature is $r = 0.694$ and without $r = 0.735$. Both prediction models have a strong Pearson correlation coefficient, indicating that the models are able to adjust the departure time estimates according to the actual departures. However, neither of the model has a very strong relationship.

As Bane NOR's model produces multiple departure estimates for each actual delay, there is an $N : 1$ relation between them. The correlation coefficient is, in other words, effectively evaluating the general trend of these estimates against the actual value. In the case of the LSTM model's 1 : 1 relationship, the correlation coefficient provides a direct measure of how well each prediction align with the corresponding actual delay. This indicates that individual errors or inaccuracies in the model's predictions directly impact the correlation coefficient.

7.3.2.2 Comparison of the Performance Metrics

The mean average error for both Bane NOR´s model and all of the LSTM model versions are below two minutes, as seen in table 6.3.2. MAE below two minutes is considered as good performance for a train delay prediction model, compared to other studies presented in table 3.5.1. Further, all four LSTM model performances are more accurate than Bane NOR´s model. With MAE equal to 1.18 minutes, the $\overline{LSTM}$ model without weather feature is the most precise prediction model. Compared to Bane NOR´s MAE, which is equal to 1.79 minutes, the model has over half a minute more average error than the $\overline{LSTM}$ model. Additionally, when comparing the root mean squared error, the prediction performance of Bane NOR's prediction model seems to have larger and less skewed errors. However, this counts mostly for the model when not excluding the bigger delays. Bane NOR´s $RMSE_{[-5.075,10.005]}$ equals 2.55 minutes while the $\overline{LSTM}$ model without weather feature equals 1.70 minutes.

Both MAE and RMSE can be interpreted directly in the units of the output variable. In the context of train delay prediction, $MAE = 1.18$ minutes indicates that, on average, the $\overline{LSTM}$ model's predictions are off by approximately 1.18 minutes, meaning that passengers can expect to actually depart within a one-minute window around the predicted time. It is worth noting, however, that this is the average error and that the actual error for a specific prediction could be both higher or lower. There is also the possibility of larger errors occasionally, which is something the RMSE can help assess. For the $\overline{LSTM}$ model, $RMSE = 1.70$ minutes indicates that the spread of the error distribution around the actual delay time is roughly 1.70 minutes. RMSE offers an understanding of the error variability, including the risk of larger errors. So, a lower RMSE not only implies a lower average error but also that large errors are less common, which in many situations, such as public transport scheduling, can be crucial.

7.3.2.3 Visual Inspection of Box Plots

The box plots give a good reflection of stations with more distributed and worse estimate deviations throughout the train route. Figure 6.3.1 shows the two models estimate deviation when including all delays. The figure illustrates how Bane NOR´s model, red boxes, have both more and bigger deviation outliers on every station, compared to the $\overline{LSTM}$ model. Even some stations have estimate deviation above 200 minutes, which is an estimate deviation above three hours. Additionally, especially the three stations *Oslo S*, *Lillehammer* and *Støren* have negative estimate deviations, below 200 minutes. This means that the train is estimated to be three hours delayed, but it reach the station on scheduled time.

When only comparing the interquartile range of each box, the two models are more comparable. Figure 6.3.2 and 6.3.4 illustrates the characteristics of the boxes for both models, illustrating the deviations with and without the deviation boundaries. Bane NOR´s model seems to have a bigger span in estimate deviation, as the interquartile range is bigger compared to the $\overline{LSTM}$ model´s, for all stations. These observations indicates that Bane NOR´s model predicts more and too optimistic estimates than the $\overline{LSTM}$ model. This optimistic trend was expected to find, as it was one of the found in the study by Skjøren (2022), as noted in 3.6. Further, the red boxes also lies above the blue ones, having a higher positive span. The median of the red boxes are always above $y = 0$, indicating an overall positive estimate deviation. On the other hand, the interquartile ranges of the blue boxes, representing the LSTM model´s estimate deviations, spans more over both sides of $y = 0$. The departure time estimate misses with both a little bit before and behind the actual departure, according to the LSTM model, which appears to have a realistic approach. Further, the majority of the blue boxes medians are below $y = 0$. This indicates that the $\overline{LSTM}$ model predicts estimated departure times before the actual departures. However, figure 6.3.4 illustrates how small estimate deviations are. There are only a few medians below -1, indicating that the rest deviation medians are less than a minute. In other words, the two models have the opposite estimate deviation median, where the Bane NOR model is too optimistic, while the $\overline{LSTM}$ model is too pessimistic.

Additionally, the exact absolute estimate deviation average median was calculated. The $\overline{LSTM}$ model´s absolute estimate deviation average median is 23.42 seconds, while Bane NOR´s models is 104.4 seconds. Further, when excluding the problematic station Oslo S, the estimate deviation average median is 76.8 seconds. These numbers indicates that the $\overline{LSTM}$ model predicts more accurate estimates all over.

Figure 6.3.3 illustrates how the estimate deviation for Bane NOR´s model on Oslo S decreases when the bigger absolute estimate deviations are removed. The median on Oslo S decreases from 8.7 minutes to 2.35 minutes. The lower boundary is ten minutes and upper is 15 minutes. Further, almost none of the other boxes seems to be touched by the restrictions. Actually, as commented in 6.3, the highest maximum estimate deviation for the $\overline{LSTM}$ model is 9.96546 minutes, on *Oslo S*, and lowest minimum is -9.070591 minutes, on *Støren*. This indicates that the model´s estimate deviations never reach outside the boundaries, and will therefor have the same absolute estimate deviation average median on 23.42 seconds. Bane NOR´s absolute estimate deviation average median is on the other hand reduced from 104.4 seconds to 84.5 seconds. Further, when excluding the problematic station Oslo S, the estimate deviation average median is 81.4 seconds seconds. These numbers indicates that the $\overline{LSTM}$ model predicts more accurate estimates all over. Additionally, Oslo S is a problematic station for Bane NOR´s prediction model, but only when including large estimate deviations.

Bane NOR´s predicted departure time seems to have an optimistic trend, as mentioned above. However, from a railway perspective, this might be a tactical decision. A train is not allowed to depart from a station before the official, published departure time. In other words, if the train is ready to depart from a station before a predicted departure time, an already delayed train has to wait, and will become even more delayed. An overly pessimistic estimate is not very approaching customer service. That means predicting departure times too late can result in worse effects than predicting them too early.

In terms of communicating train delays to the end-users, there are a number of ways this could be done. A direct delay prediction provides a simple, easily understandable figure that passengers can use to adjust their plans. It does not, however, represent the uncertainty inherent in any prediction. If the actual delay significantly deviates from this single value, it could lead to dissatisfaction among passengers. A more nuanced approach might involve providing a delay prediction along with a confidence interval. Rather than giving a single value, this would present a range within which the actual delay is likely to fall, offering a clearer picture of the uncertainty involved. Alternatively, the delay prediction could be accompanied with a probability, indicating a confidence level of the prediction. This could quantify the certainty of the prediction, helping passengers manage expectations. The downside of these additional degrees of information is that they could serve to over-complicate, and confuse the passengers.

Adding a buffer to the predicted delay could also enhance the utility of delay predictions. This buffer would serve as an additional margin of error, further ensuring that passengers have sufficient time to accommodate any unpredicted variations in train delays. It could be beneficial, as the cost of underestimating a delay is often higher than the cost of overestimating it. However, consistently overestimating delays might lead to unnecessary wait times for passengers. Also, if the buffer is perceived as an arbitrary addition to predicted delays, it could lead to skepticism about the precision of the prediction model. Thus, if implemented, it should be clearly communicated as a precautionary measure rather than a correction for prediction inaccuracies.

7.4 Limitations

The data used in this study was received in three separate data sets, from two different sources, further described in section 4. There were two data sets from TIOS database, one with raw data and the other one with pre-processed data. Further, one data set with historical daily meteorological observations was gathered from the Norwegian Meteorological Institute. Both historical train observation data sets from TIOS included noisy data with outliers. Therefore, abnormal data has been removed, including duplicates, invalid data, and incorrect data.

The process of acquiring weather data from Met Norway presented a set of challenges. The data for specific train station locations subject to analysis was not consistently available. Some stations lacked certain weather elements altogether, while others had data for only certain time periods. Furthermore, these time periods did not always align with the time interval pertinent to this analysis.

As detailed in section 5.2.1.3, data imputation techniques were employed to counteract the potential adverse effects of these missing values. In instances where data were unavailable, we substituted values from nearby stations or calculated and used mean values from multiple proximate stations.

Despite these efforts, the data imputation approach failed to address all instances of missing data, which left some data points blank. This led to the exclusion of entire rows of data. Should these instances of missing data not be randomly distributed but part of a discernible pattern, they could introduce bias, thereby rendering the weather features as noise. This potential bias could offer an explanation for the inferior performance of the model that included weather data compared to the model that did not incorporate such data. The missing weather data could, for instance, be related to other factors such as certain weather conditions leading to missing data due to equipment failure. The missing data would not be a random subset of the weather data, but a subset related to the very conditions which we aim to model, leading to a biased and potentially misleading model.

When comparing the two models, the data is imbalanced as a result of the models' different modelling paradigms. Since Bane NOR´s model predicts many estimates for each station on a train route, the data size is bigger than for the LSTM model. When comparing the two data sets used for model evaluation, Bane NOR´s model has measured 355.281 unique actual delays and estimated delay, while the LSTM model has 13.162 ones. This results in imbalanced data when comparing the two data sets, indicating reflection of an unequal distribution. However, when comparing the two data sets, box plots are used. The box plot is a good trade-off between summary statistics and data visualization. As long as both data set has enough data to be representative, the two different data sizes will not have too much of an impact on the analysis. An issue with the box plot is that it hides the shape of the data, telling us some summary statistics but not showing us the actual data distribution. However, when comparing the overall performance of the models and their estimate deviation, a statistic summary gives a reflective overview.

The weather data set and the TIOS pre-processed data set both include data that covers the period from 01.01.21 to 28.02.23, just above two years. This period was impacted by COVID-19 pandemic. It has been demonstrated in other research that the pandemic affected peoples' travel behaviors, which is likely to have an influence on our data sets. However, the pandemic seems to have changed peoples travelling patterns, which indicates that data before the pandemic might be outdated. It is challenging to draw any conclusions when projecting travel patterns, but this supports the case for dynamic models that can be continually improved and adapted to new patterns by using new training data. This is an advantage of the machine learning model, as explained above.

TIOS raw data spans from the summer, 15th of May 2022, until the end of the winter, 22th of March 2023. Compared to the two other data sets, the period is smaller. The data might therefore include abnormalities that have occurred in the same period, or not being able to observe a travel pattern. The quantity of the data after data wrangling was 2.715.816 train records, which is considered enough data to allow for a model analysis that was representative of the data.

The paper only examines the long-distance train route, running between the capital *Oslo* and *Trondheim.* Additionally, the rail network consists primarily of single-track lines with little traffic. More traffic and multiple-track lines are accessible the closer you are to Oslo S. The performance of the LSTM model as well as the training and testing processes will be impacted by these railway factors. Other patterns is expected to be observed when analysing other train routes. The different features of the model might then have a bigger impact on the prediction outcomes, like day of the week and week of the year. Further, as the train route has less traffic, a train is more rarely affected by others in the rail network. Other train routes might experience more delay propagation, leading to more noisy data. This limitation also counts when examining the prediction model used by Bane NOR. The event-driven model based on a linear equation system is generic with the same framework regardless of the rail behaviour patterns. It is therefore assumed that when examining the models performance on other routes, one will get another result.

One of the primary objectives of this master's thesis was to develop an LSTM train delay prediction model, and to compare it to Bane NOR's existing prediction model. As presented throughout this report, the two models differ greatly both with regard to their respective modelling paradigm, data utilisation, and prediction horizons. In order to facilitate a comparison between the two models, a selective analysis of Bane NOR's model has been conducted. This mainly includes narrowing the prediction horizon and setting outlier boundaries similar to those used for the LSTM model. While some might argue that the models are incomparable by nature, it could be argued that this selective analysis of Bane NOR's model has contributed to reducing the gap between the differences. Only accounting for estimates, generated by Bane NOR's model, in a window of $[50, 5]$ minutes before the departure at a station, has helped approximate the prediction procedure of the LSTM model, which only predicts delays at the next station. However, the estimates generated is purely based on linear functions. By training on historical train data, the LSTM works as a complex function approximator, creating a mapping between input variables and the target variable. Hence, the LSTM serves as a more powerful and complex tool accounting for numerous factors contributing to the train delays.

While the evaluation of the LSTM model has shown promising results in terms of MAE and RMSE, it is crucial to consider the treatment of outliers in the training data when interpreting these results. The outlier boundaries $[-5.075, 10.005]$ were set to focus the model on the most common delays, and to minimize the impact of very large delays that could disproportionately influence the model's learning. However, this implies that delays within this range are the ones the model has primarily learned to predict. The boundary limitation inherently reduces the potential magnitude of any individual error. For instance, the greatest possible absolute error would be 15 minutes if a delay of 10 minutes was predicted as -5 minutes. Consequently, the MAE and RMSE, being averages of these individual errors, are naturally bound to be relatively low due to the restricted range of possible errors. This is an important consideration for the broader application of the model. Potential implementation in a real-world scenario containing larger delays would yield lower performance.

Moreover, the assessment of the LSTM model's performance bears certain constraints. Prior to the model's construction and training phase, the data was partitioned into training, validation, and test sets. Due to the sequential nature of the data and the historical delay patterns embedded within, these sets were split chronologically rather than randomly shuffled. Consequently, the training set comprises the initial 70% of the data, while the validation and test sets account for the subsequent 15% each. As a result, the test set, which is used to calculate the performance metrics of the model, encompasses data from late November 2022 to the end of February 2023. This confines the performance evaluation to a relatively narrow time period, which may not fully represent the model's overarching performance.

One approach to account for this limitation and gain a more comprehensive understanding of a machine learning model's performance is through Cross-Validation. This technique involves dividing the data into multiple subsets or *folds*. The model is then trained and tested multiple times, each time using a different fold as the test set and the remaining folds as the training set. This ensures that the model's performance is evaluated on various parts of the data, providing a more holistic assessment. Moreover, the results obtained from cross-validation could help in tuning the model's hyper-parameters and potentially improve its predictive power, thereby enhancing its utility in real-world applications. As stated in section 3.2.1, Oneto et al. (2018) developed a neural network train delay prediction model which outperformed traditional models. The study demonstrated how exploitation of the Cross-Validation technique contributed to improved performance through optimal hyper-parameter tuning.

Finally, it is essential to mention that the time frame of the LSTM model evaluation is a subset of the time period during which Bane NOR's model was analysed. Bane NOR's model was assessed on data from the 15th of June 2022 to the 22nd of March 2023. This discrepancy in evaluation periods has implications for the comparative analysis between the two models. The broader time frame for Bane NOR's model means that it might be influenced by factors that the LSTM model did not have the opportunity to encounter in its test set. On the positive side, despite the discrepancy in evaluation periods, there is overlap in the timeframes that the two models were evaluated on. This shared subset of data allows for some degree of comparability between the models. This offers a valuable opportunity to directly compare how each model handles the same set of circumstances, and it strengthens the validity of the comparative analysis conducted in this study.

CHAPTER EIGHT

CONCLUSIONS

This study embarked on a quest to explore the performance of two different prediction models for estimating train departure delays: a machine learning model utilising Long Short-Term Memory (LSTM) networks, and Bane NOR's current model. The principal objective was to determine if machine learning, particularly LSTM, could provide a more accurate prediction model, especially under fluctuating weather conditions.

Our first exploration was focused on **RQ 1**, which concerned the utilization of LSTM models in predicting small, everyday train delays. The LSTM model was developed and tested with and without the inclusion of weather features. The key performance metrics such as the Pearson Correlation Coefficient, Mean Average Error (MAE), and Root Mean Squared Error (RMSE) were used to assess and compare the performance of both models. Through this, we were able to establish that LSTM models are powerful in capturing delay patterns owing to their ability to retain and process historical data over long sequences.

Following this, we addressed **RQ 2**, which explored how various types of train-related features contribute to an LSTM prediction model's performance. The results revealed that certain features, specifically the time-lags of the departure delay and the delay classification feature, had a significant positive impact on the model's predictive accuracy.

Subsequently, we turned our attention to **RQ 3**, which focused on the impact of incorporating weather data as an external feature on the predictive accuracy of LSTM models for train delay predictions. While the expectation was that weather features would enhance the model's performance, the results did not evidence an improvement. However, it is worth noting that the lack of improvement could be attributed to data limitations encountered, such as missing weather data and potential biases introduced during the data imputation process. This may have obfuscated the true influence of weather conditions on train departure delay, and could be an area for future investigation.

Finally, we addressed **RQ 4**, conducting a comparative analysis of the LSTM model and Bane NOR's current prediction model. Despite the data challenges and the specific focus on one train route in Norway, the LSTM model demonstrated a stronger correlation and lower error rates than the existing model.

While this study focused on a specific train route in Norway, the findings bear wider implications for the transport industry. Machine learning models, such as LSTM, have the potential to enhance the accuracy of predictive models, thereby improving operational efficiency and customer satisfaction. These models are adaptable, and can be continually improved and refined with new data, making them a robust solution in a rapidly changing world.

Despite some limitations, such as imbalanced data sets, data quality issues, and potential seasonal biases in the performance evaluation, this research contributes to a growing body of literature advocating for the use of machine learning in transportation modelling. Future research should focus on improving data quality, testing the models on various train routes with different characteristics, and possibly incorporating more sophisticated techniques such as cross-validation for a more robust performance assessment.

In conclusion, this study provides evidence that machine learning, and LSTM models in particular, have the potential to outperform traditional methods in predicting train departure times. The results emphasize the value of a comprehensive approach to train delay prediction, one that integrates weather data, exploits historical delay patterns, and incorporates the influence of previous station delays. This strategy, coupled with careful pre-processing and model tuning all contributed to the development of a model that could effectively handle the complexity and variability inherent in train delay data. Future research can build upon these findings, exploring other feature engineering techniques or external factors that might enhance the model's predictive power even further. As such, the LSTM model could serve as a valuable tool in improving the efficiency and reliability of railway operations.

8.1 Future work

The insights derived from addressing the research questions offer a sound foundation upon which further advancements can be made. Several areas of focus for future work, in light of the research conducted, are as follows:

Real-time Predictions While this study used historical data for predictions, future research could also look at enabling the LSTM model to make real-time delay predictions by utilising real-time data, enhancing its practical utility for railway operation management. Section 7.3.1 offered two potential ways of doing this. The first option is continuously fine-tuning the pre-trained LSTM model with new incoming real-time data. This approach depends on the correct balance between the model retaining past knowledge learned from historical data, and adapting to new patterns provided by the real-time data. The second option includes real-time predictions for multiple stations. This could be done transforming the predictive LSTM model to a generative LSTM model, generating a sequence of predicted delays for multiple future stations. As the train progresses along its route, and actual real-time delay data becomes available, this data is used to update the input sequence.

Impact of Weather Conditions: Despite the lack of improvement in model performance with the inclusion of weather features in this study, further research on how weather conditions impact train delays is recommended. This holds with regard to predictions of both minor and major delays. Enhancing the granularity and quality of weather data, particularly concerning severe weather conditions that lead to more substantial delays, could offer more nuanced insights and improve the model's predictive performance.

Inclusion of Exogenous Information: The findings concerning RQ2 and RQ3 highlight the potential value of additional features in enhancing the predictive performance of the LSTM model. Future research could involve the analysis and inclusion of further exogenous information affecting railway operations, such as real-time passenger load, track maintenance schedules, and special events, among others. Such information, if available, could provide a richer feature set for the LSTM model, potentially leading to more accurate delay predictions.

Examination Across Various Rail Lines: This study's findings are based on a single long-distance railway route in Norway. Given the uniqueness of each rail line, future work should focus on testing the LSTM model on different sections of the railway network. Such an examination would help validate the model's adaptability and effectiveness across diverse operational and environmental conditions.

Cross Validation: The implementation of more robust techniques such as Cross-Validation can be beneficial for enhancing the generalisability and robustness of the LSTM model's performance. Cross-validation, which involves partitioning the data set into multiple subsets and validating the model on each, could provide a more thorough performance assessment and facilitate model refinement. Also, a more comprehensive approach to the Grid Search method, described in section 7.1.3, could provide further refinements of the hyper-parameters of the model. Consequently, this could further improve model performance.

Delay Communication: Future work in the area of delay communication can explore strategies that balance the simplicity and comprehensibility of direct delay predictions with the detailed information provided by techniques like confidence intervals or probability measures. The impact of these strategies on passenger satisfaction, perception of prediction accuracy, and overall travel experience could be studied to inform the development of more effective communication methods.

Buffer: Future research could focus on testing the utility of incorporating a buffer into delay predictions. This would involve determining optimal buffer sizes that balance accuracy, manage unforeseen delay variations, and maintain passenger trust in the prediction model.

In summary, the research insights gleaned from this study offer a platform for further exploration and advancement in the field of machine learning-based train delay prediction. By continuing to build upon these findings, it is anticipated that future research will contribute to enhancing the precision and reliability of train delay predictions, ultimately leading to improved railway operations.

REFERENCES

Abril, Montserrat et al. (2008). ‘An assessment of railway capacity’. In: *Transportation Research Part E: Logistics and Transportation Review* 44.5, pp. 774–806.

Adjetey-Bahun, Kpotissan et al. (2016). ‘A model to quantify the resilience of mass railway transportation systems’. In: *Reliability Engineering & System Safety* 153, pp. 1–14.

Andersson, Emma V, Anders Peterson and Johanna Törnquist Krasemann (2015). ‘Reduced railway traffic delays using a MILP approach to increase Robustness in Critical Points’. In: *Journal of Rail Transport Planning & Management* 5.3, pp. 110–127.

Baker, CJ et al. (2010). ‘Climate change and the railway industry: a review’. In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 224.3, pp. 519–528.

Bane NOR (2023a). *Sjekklister Togleder - 7 Beredskap: Gul, oransje og rød.* https://orv.banenor.no/beredskap/doku.php. Accessed: 2023-04-16.

— (2022). *Tele/Prosjektering og bygging/Informasjonssystemer.* https://trv.banenor.no/wiki/Tele/Prosjektering_og_bygging/Informasjonssystemer.

— (2023b). $banenettverk_{n}etwork_{s}tatement_{s}trekningskart_{s}ide_{2}.png$. URL: https://networkstatement.banenor.no/lib/exe/detail.php?id=vedlegg%5C%3Astrekningskart&media=vedlegg:banenettverk_network_statement_strekningskart_side_2.png (visited on 19/04/2023).

— (2023c). $dovb_{r}aub_{r}osb_{s}ide5.png$. URL: https://networkstatement.banenor.no/lib/exe/detail.php?id=vedlegg%5C%3Astrekningskart&media=vedlegg:dovb_raub_rosb_side5.png (visited on 19/04/2023).

— (2023d). *Line Maps.* URL: https://networkstatement.banenor.no/doku.php?id=vedlegg:strekningskart (visited on 19/04/2023).

— (2023e). *Punktlighetsrapport 2021.* URL: https://www.banenor.no/contentassets/a4687a01d295498ca3fa0a7ef71e3102/punktlighetsrapport-2021.pdf (visited on 09/03/2023).

Banerjee, Nilabhra, Alec Morton and Kerem Akartunalı (2020). ‘Passenger demand forecasting in scheduled transportation’. In: *European Journal of Operational Research* 286.3, pp. 797–810.

Banko, Michele and Eric Brill (2001). 'Scaling to very very large corpora for natural language disambiguation'. In: *Proceedings of the 39th annual meeting of the Association for Computational Linguistics*, pp. 26–33.

Bengio, Y., P. Frasconi and P. Simard (1993). 'The problem of learning long-term dependencies in recurrent networks'. In: *IEEE International Conference on Neural Networks*, 1183–1188 vol.3. DOI: 10.1109/ICNN.1993.298725.

Berger, Annabell et al. (2011). 'Stochastic delay prediction in large train networks'. In: *11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Bergstra, James and Yoshua Bengio (2012). 'Random search for hyper-parameter optimization.' In: *Journal of machine learning research* 13.2.

Bishop, Chris M (1994). 'Neural networks and their applications'. In: *Review of scientific instruments* 65.6, pp. 1803–1832.

Bisong, Ekaba (2019). 'Introduction to Scikit-learn'. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, pp. 215–229.

Blayac, Thierry and Maıté Stéphan (2021). 'Are retrospective rail punctuality indicators useful? Evidence from users perceptions'. In: *Transportation Research Part A: Policy and Practice* 146, pp. 193–213.

Borndörfer, Ralf et al. (2018). *Handbook of Optimization in the Railway Industry*. Springer.

Bostrom, Nick and Eliezer Yudkowsky (2018). 'The ethics of artificial intelligence'. In: *Artificial intelligence safety and security*. Chapman and Hall/CRC, pp. 57–69.

Bussieck, Michael (1998). 'Optimal lines in public rail transport'. PhD thesis. Univ.-Bibl.

Bussieck, Michael R, Thomas Winter and Uwe T Zimmermann (1997). 'Discrete Optimization in Public Rail Transport'. In: *Mathematical Programming* 79, pp. 415–444.

Cacchiani, Valentina, Dennis Huisman et al. (2014). 'An overview of recovery models and algorithms for real-time railway rescheduling'. In: *Transportation Research Part B: Methodological* 63, pp. 15–37.

Cacchiani, Valentina and Paolo Toth (2012). 'Nominal and robust train timetabling problems'. In: *European Journal of Operational Research* 219.3, pp. 727–737.

Caprara, Alberto, Matteo Fischetti and Paolo Toth (2002). 'Modeling and solving the train timetabling problem'. In: *Operations research* 50.5, pp. 851–861.

Cerreto, Fabrizio et al. (2016a). 'Causal analysis of railway running delays'. In: *11th World Congress on Railway Research (WCRR 2016), Milan, Italy.*

— (2016b). 'Causal analysis of railway running delays'. In: *11th World Congress on Railway Research (WCRR 2016), Milan, Italy.*

Chapman, Lee (2007). 'Transport and climate change: a review'. In: *Journal of transport geography* 15.5, pp. 354–367.

Chen, James (2023). *How a Histogram Works to Display Data*. URL: https://www.investopedia.com/terms/h/histogram.asp (visited on 26/04/2023).

Chen, Tianqi and Carlos Guestrin (2016). 'XGBoost: A Scalable Tree Boosting System'. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, pp. 785–794. ISBN: 9781450342322. DOI: 10.1145/2939672.2939785. URL: https://doi.org/10.1145/2939672.2939785.

Chung, Junyoung et al. (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

Connor, Jerome T, R Douglas Martin and Les E Atlas (1994). 'Recurrent neural networks and robust time series prediction'. In: *IEEE transactions on neural networks* 5.2, pp. 240–254.

Conte, Carla and Anita Schöbel (2007). 'Identifying dependencies among delays'. In: *Proceedings of IAROR* 2007.

Cordeau, Jean-François, Paolo Toth and Daniele Vigo (1998). 'A survey of Optimization Models for Train Routing and Scheduling'. In: *Transportation Science* 32, pp. 380–404.

Corman, Francesco and Jonas Henken (2022). 'Estimating aggregate railway performance from realized empirical data: Literature review, a test case and a research roadmap'. In: *Journal of Rail Transport Planning & Management* 22, p. 100316.

Corman, Francesco and Pavle Kecman (2018). 'Stochastic prediction of train delays in real-time using Bayesian networks'. In: *Transportation Research Part C: Emerging Technologies* 95, pp. 599–615.

Corman, Francesco and Lingyun Meng (2015). 'A Review of Online Dynamic Models and Algorithms for Railway Traffic Management'. In: *IEEE Transactions on Intelligent Transportation Systems* 16.3, pp. 1274–1284. DOI: 10.1109/TITS.2014.2358392.

Corman, Francesco, Dario Pacciarelli et al. (2015). 'Rescheduling railway traffic taking into account minimization of passengers' discomfort'. In: *Computational Logistics: 6th International Conference, ICCL 2015, Delft, The Netherlands, September 23-25, 2015, Proceedings 6*. Springer, pp. 602–616.

D'Ariano, Andrea and Marco Pranzo (2009). 'An advanced real-time train dispatching system for minimizing the propagation of delays in a dispatching area under severe disturbances'. In: *Networks and Spatial Economics* 9, pp. 63–84.

Dekker, Mark M et al. (2019). 'Predicting transitions across macroscopic states for railway systems'. In: *PloS one* 14.6, e0217710.

Dewilde, Thijs (2014). 'Improving the robustness of a railway system in large and complex station areas'. In.

Dey, Rahul and Fathi M Salem (2017). 'Gate-variants of gated recurrent unit (GRU) neural networks'. In: *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*. IEEE, pp. 1597–1600.

Dong, Guozhu and Huan Liu (2018). *Feature engineering for machine learning and data analytics*. CRC Press.

Enquiries, National Rail (2023). *Leaf Fall Timetable Changes - Autumn 2022.* URL: https://www.nationalrail.co.uk/255905.aspx (visited on 14/03/2023).

Europe´s Rail (2020). *Shift2Rail.* https://rail-research.europa.eu. Accessed: 2022-10-18.

European Commission (2021). *Europe´s Rail Joint Undertaking.* https://rail-research.europa.eu/wp-content/uploads/2022/03/EURAIL_Master-Plan.pdf. Accessed: 2022-10-18.

Evans, James D (1996). *Straightforward statistics for the behavioral sciences.* Thomson Brooks/Cole Publishing Co.

Faverges, Marie Milliet de et al. (2018). 'Estimating long-term delay risk with Generalized Linear Models'. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC).* IEEE, pp. 2911–2916.

Feurer, Matthias and Frank Hutter (2019). 'Hyperparameter optimization'. In: *Automated machine learning: Methods, systems, challenges*, pp. 3–33.

Filcek, Grzegorz et al. (2021). 'An Algorithm for Rescheduling of Trains under Planned Track Closures'. In: *Transportation Research Part B: Methodological* 63, pp. 15–37.

Galarnyk, Michael (2022). *Understanding Boxplots.* URL: https://builtin.com/data-science/boxplot (visited on 14/10/2022).

Gestrelius, Sara, Markus Bohlin and Martin Aronsson (2015). 'On the uniqueness of operation days and delivery commitment generation for train timetables'. In: *6th International Conference on Railway Operations Modelling and Analysis (RailTokyo2015), March 23-26, 2015, Tokyo, Japan.*

Ghofrani, Faeze et al. (2018). 'Recent applications of big data analytics in railway transportation systems: A survey'. In: *Transportation Research Part C: Emerging Technologies* 90, pp. 226–246.

Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep learning.* MIT press, pp. 367–415.

Google (2023). *Google Maps Geocoding API.* URL: https://developers.google.com/maps/documentation/geocoding (visited on 09/02/2023).

Goverde, Rob MP (2005a). 'Punctuality of railway operations and timetable stability analysis'. In.

— (2005b). 'Punctuality of railway operations and timetable stability analysis'. In.

— (2007). 'Railway timetable stability analysis using max-plus system theory'. In: *Transportation Research Part B: Methodological* 41.2, pp. 179–201.

Goverde, Rob MP and Ingo A Hansen (2013). 'Performance indicators for railway timetables'. In: *2013 IEEE International Conference on intelligent rail transportation proceedings.* IEEE, pp. 301–306.

— (2000). 'TNV-Prepare: Analysis of Dutch railway operations based on train detection data'. In: *Computers in Railways* 7, pp. 779–788.

Group, Funkwerk (2023). *Funkwerk mobility platform: our references. BANE NOR – NORWAY.* URL: https://funkwerk-mobility-platform.com/en/referenz-bane-nor/ (visited on 19/04/2023).

Handelman, Guy S et al. (2019). 'Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods'. In: *American Journal of Roentgenology* 212.1, pp. 38–43.

Harris, Nigel G, Christian S Mjøsund and Hans Haugland (2013). 'Improving railway performance in Norway'. In: *Journal of Rail Transport Planning & Management* 3.4, pp. 172–180.

Harrod, Steven S (2012a). 'A tutorial on fundamental model structures for railway timetable optimization'. In: *Surveys in Operations Research and Management Science* 17.2, pp. 85–96.

— (2012b). 'A tutorial on fundamental model structures for railway timetable optimization'. In: *Surveys in Operations Research and Management Science* 17.2, pp. 85–96.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780.

Holloway, Catherine et al. (2016). 'Effect of vertical step height on boarding and alighting time of train passengers'. In: *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit* 230.4, pp. 1234–1241.

Hong, Liu et al. (2019). 'Spatiotemporal vulnerability analysis of railway systems with heterogeneous train flows'. In: *Transportation Research Part A: Policy and Practice* 130, pp. 725–744.

Hopfield, John Joseph (1982). 'Neural networks and physical systems with emergent collective computational abilities'. In: *Proceedings of the National Academy of Sciences of the United States of America* 79.8, pp. 2554–2558. DOI: https://doi.org/10.1073/pnas.79.8.2554.

Hornik, Kurt, Maxwell Stinchcombe and Halbert White (1990). 'Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks'. In: *Neural networks* 3.5, pp. 551–560.

Huang, Ping, Zhongcan Li et al. (2021). 'Modeling train timetables as images: A cost-sensitive deep learning framework for delay propagation pattern recognition'. In: *Expert Systems with Applications* 177, p. 114996.

Huang, Ping, Chao Wen et al. (2020). 'Modeling train operation as sequences: A study of delay prediction with operation and weather data'. In: *Transportation research part E: logistics and transportation review* 141, p. 102022.

informs (2022). *Designing real-time traffic management for an urban rail transit system – London's new Elizabeth line, UK.* URL: https://connect.informs.org/railway-applications/new-item3/problem-solving-competition681 (visited on 15/10/2022).

Institute, Norwegian Meteorological (2023a). *Free meteorological data.* URL: https://www.met.no/en/free-meteorological-data (visited on 09/02/2023).

— (2023b). *Norwegian Centre For Climate Services.* URL: https://seklima.met.no/ (visited on 19/02/2023).

Jain, Abhinav et al. (2020). 'Overview and importance of data quality for machine learning tasks'. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 3561–3562.

Jernbanedirektoratet (2023). *Jernbanenettet i Norge.* URL: https://www.jernbanedirektoratet.no/no/jernbanesektoren/jernbanenettet-i-norge/ (visited on 15/02/2023).

JJ (2022). *MAE and RMSE — Which Metric is Better?* URL: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better-e60ac3bde13d (visited on 15/10/2022).

Joborn, Martin and Zohreh Ranjbar (2022). 'Understanding causes of unpunctual trains: Delay contribution and critical disturbances'. In: *Journal of Rail Transport Planning & Management* 23, p. 100339.

Jordan, Michael I and Tom M Mitchell (2015). 'Machine learning: Trends, perspectives, and prospects'. In: *Science* 349.6245, pp. 255–260.

Kauppi, Arvid et al. (2017). 'Future train traffic control: control by re-planning'. In: *Rail Human Factors.* Routledge, pp. 296–305.

Kecman, Pavle and Rob MP Goverde (2015). 'Predictive modelling of running and dwell times in railway traffic'. In: *Public Transport* 7.3, pp. 295–319.

Kjøsnes, Gunhild (2015). 'Suksessfaktorer ved utbygging av små byggeprosjekter for Jernbaneverket i en kompleks kontekst, illustrert i prosjektet om målestasjoner for værdata'. MA thesis. NTNU.

Koetse, Mark J and Piet Rietveld (2009a). 'The impact of climate change and weather on transport: An overview of empirical findings'. In: *Transportation Research Part D: Transport and Environment* 14.3, pp. 205–221.

— (2009b). 'The impact of climate change and weather on transport: An overview of empirical findings'. In: *Transportation Research Part D: Transport and Environment* 14.3, pp. 205–221.

Krueger, Harald (1999). 'Parametric modeling in rail capacity planning'. In: *Proceedings of the 31st conference on Winter simulation: Simulation—a bridge to the future-Volume 2*, pp. 1194–1200.

Lam, Terence C and Kenneth A Small (2001a). 'The value of time and reliability: measurement from a value pricing experiment'. In: *Transportation Research Part E: Logistics and Transportation Review* 37, pp. 231–251.

— (2001b). 'The value of time and reliability: measurement from a value pricing experiment'. In: *Transportation Research Part E: Logistics and Transportation Review* 38, pp. 231–251.

Langeland, Ove, Magnus Andersson and Bjørg Langset Flotve (2021). *Changes and Challenges in Future Transport: Drivers and Trends.* 5600-DIGMOB.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton (2015). 'Deep learning'. In: *Nature* 521, pp. 436–444. DOI: https://doi.org/10.1038/nature14539.

Lee, Wei-Hsun, Li-Hsien Yen and Chien-Ming Chou (2016). 'A delay root cause discovery and timetable adjustment model for enhancing the punctuality of railway services'. In: *Transportation Research Part C: Emerging Technologies* 73, pp. 49–64.

Leutwiler, Florin and Francesco Corman (2022). 'A logic-based Benders decomposition for microscopic railway timetable planning'. In: *European Journal of Operational Research* 303.2, pp. 525–540.

Li, Dewei, Winnie Daamen and Rob MP Goverde (2016). ‘Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station’. In: *Journal of Advanced Transportation* 50.5, pp. 877–896.

Li, Zheng, David A Hensher and John M Rose (2009). ‘Willingness to pay for travel time reliability in passenger transport: a review and some new empirical evidence.’ In: *Transportation Research Part E: Logistics and Transportation Review* 46, pp. 384–403.

Li, ZhongCan et al. (2021). ‘Near-term train delay prediction in the Dutch railways network’. In: *International Journal of Rail Transportation* 9.6, pp. 520–539.

Li, Zhongcan et al. (2022). ‘Prediction of train arrival delays considering route conflicts at multi-line stations’. In: *Transportation Research Part C: Emerging Technologies* 138, p. 103606.

Lillicrap, Timothy P and Adam Santoro (2019). ‘Backpropagation through time and the brain’. In: *Current opinion in neurobiology* 55, pp. 82–89.

Ling, Ximan et al. (2018). ‘Uncovering correlation between train delay and train exposure to bad weather’. In: *Physica A: Statistical Mechanics and its Applications* 512, pp. 1152–1159.

Ludvigsen, Johanna and Ronny Klæboe (2014). ‘Extreme weather impacts on freight railways in Europe’. In: *Natural hazards* 70, pp. 767–787.

Lusby, Richard M, Jesper Larsen and Simon Bull (2018a). ‘A survey on robustness in railway planning’. In: *European Journal of Operational Research* 266.1, pp. 1–15.

— (2018b). ‘A survey on robustness in railway planning’. In: *European Journal of Operational Research* 266.1, pp. 1–15.

Lüthi, Marco, Giorgio Medeossi and Andrew Nash (2007). ‘Evaluation of an integrated real-time rescheduling and train control system for heavily used areas’. In: *International Seminar on Railway Operations Modelling and Analysis (IAROR) 2007 Conference, Hannover, Germany.*

Maguire, Heather (2007). ‘Book review: Data quality: concepts, methodologies and techniques by C. Batini and M. Scannapieco’. In: *International Journal of Information Quality* 1.4, pp. 444–450.

Mannino, Carlo and Andreas Nakkerud (2023). ‘Optimal Train Rescheduling in Oslo Central Station’. In: *Omega* 116.C.

Martinelli, David R and Hualiang Teng (1996). ‘Optimization of railway operations using neural networks’. In: *Transportation Research Part C: Emerging Technologies* 4.1, pp. 33–49.

Mascis, Alessandro and Dario Pacciarelli (2002). ‘Job-shop scheduling with blocking and no-wait constraints’. In: *European Journal of Operational Research* 143.3, pp. 498–517.

Nagy, Enikő and Csaba Csiszár (2015). ‘Analysis of delay causes in railway passenger transportation’. In: *Periodica Polytechnica Transportation Engineering* 43.2, pp. 73–80.

Nair, Rahul et al. (2019). ‘An ensemble prediction model for train delays’. In: *Transportation Research Part C: Emerging Technologies* 104, pp. 196–209.

Narayanaswami, Sundaravalli and Narayan Rangaraj (2011). ‘Scheduling and rescheduling of railway operations: A review and expository analysis’. In: *Technology Operation Management* 2, pp. 102–122.

Norheim, B. and A. Ruud (2001). ‘The value of time and reliability: measurement from a value pricing experiment’. In: *Transportation Research Part E: Logistics and Transportation Review* 38, pp. 231–251.

Nyström, Birre (June 2008). ‘Aspects of improving punctuality: from data to decision in railway maintenance’. PhD thesis. Luleå University of Technology.

Økland, Andreas and Nils Olsson (2021). ‘Punctuality development and delay explanation factors on Norwegian railways in the period 2005–2014’. In: *Public Transport* 13.1, pp. 127–161.

Olsson, Nils, Askill Harkjerr Halse et al. (2015). *Punktlighet i jernbanen-hvert sekund teller.*

Olsson, Nils and Hans Haugland (2004). ‘Influencing factors on train punctuality—results from some Norwegian studies’. In: *Transport policy* 11, pp. 387–397.

Oneto, Luca et al. (2016). ‘Advanced analytics for train delay prediction systems by including exogenous weather data’. In: *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, pp. 458–467.

— (2017). ‘Dynamic delay predictions for large-scale railway networks: Deep and shallow extreme learning machines tuned via thresholdout’. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 47.10, pp. 2754–2767.

— (2018). ‘Train delay prediction systems: a big data analytics perspective’. In: *Big data research* 11, pp. 54–64.

Palmqvist, Carl-William (2019). ‘Delays and timetabling for passenger trains’. PhD thesis. Lund University.

Palmqvist, Carl-William and Ida Kristoffersson (2022). ‘A Methodology for Monitoring Rail Punctuality Improvements’. In: *IEEE Open Journal of Intelligent Transportation Systems* 3, pp. 388–396.

Palmqvist, Carl-William, Nils OE Olsson and Lena Winslott Hiselius (2017). ‘Some influencing factors for passenger train punctuality in Sweden’. In: *International Journal of Prognostics and Health Management* 8.3.

Parbo, J., J.O. Nielsen and C.G. Prato (2015). ‘Passenger Perspectives in Railway Timetabling: A Literature Review’. In: *Transport Reviews* 36, pp. 500–526.

Perkins, Adam, Brendan Ryan and Peer-Olaf Siebers (2015). ‘Modelling and simulation of rail passengers to evaluate methods to reduce dwell times’. In.

Petropoulos, Fotios et al. (2022). ‘Forecasting: theory and practice’. In: *International Journal of Forecasting.*

Pham, Binh Thai, Dieu Tien Bui and Indra Prakash (2017). ‘Landslide Susceptibility Assessment Using Bagging Ensemble Based Alternating Decision Trees, Logistic Regression and J48 Decision Trees Methods: A Comparative Study’. eng. In: *Geotechnical and geological engineering* 35.6, pp. 2597–2611. ISSN: 0960-3182.

Profillidis, V.A. (2006). *Railway Management and Engineering.* Farnham, Surrey, England: Ashgate Publishing.

PyTorch (2023). *LSTM.* URL: https://pytorch.org/docs/stable/generated/torch.nn.LSTM.html (visited on 23/05/2023).

Rattenbury, Tye et al. (2017). *Principles of data wrangling: Practical techniques for data preparation.* " O'Reilly Media, Inc."

Rodriguez-Galiano, V et al. (2015). 'Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines'. In: *Ore Geology Reviews* 71, pp. 804–818.

Rosenblatt, Frank (1958). 'The perceptron: A probabilistic model for information storage and organization in the brain.' In: *Psychological Review* 65.6, pp. 386–408. DOI: https://doi.org/10.1037/h0042519.

Rudnicki, A. (1997). 'Measures of regularity and punctuality in public transport'. In: *IFAC Proceedings Volumes* 30, pp. 661–666.

Rumelhart, David E, Geoffrey E Hinton and Ronald J Williams (1986). 'Learning representations by back-propagating errors'. In: *nature* 323.6088, pp. 533–536.

Russell, Stuart J and Peter Norvig (2010a). *Artificial intelligence a modern approach.* Pearson Education, Inc.

— (2010b). *Artificial intelligence a modern approach.* Pearson Education, Inc., pp. 1034–1040.

Sabir, Muhammad (2011). 'Weather and travel behaviour'. PhD thesis. Vrije Universiteit Amsterdam.

Şahin, İsmail (2017). 'Markov chain model for delay distribution in train schedules: Assessing the effectiveness of time allowances'. In: *Journal of Rail Transport Planning Management* 7, pp. 101–117.

Sameni, Melody Khadem, Alex Landex and John Preston (2011). 'Developing the UIC 406 method for capacity analysis'. In: *4th international seminar on railway operations research.* Italy Rome.

Sameni, Melody Khadem and Arash Moradi (2022). 'Railway capacity: A review of analysis methods'. In: *Journal of Rail Transport Planning & Management* 24, p. 100357.

Schlechte, Thomas, Ralf Borndörfer and Martin Grötschel (2007). 'Models for Railway Track Allocation'. In.

Sentralbyrå, Statistics (2023). *13482: Gods- og persontransport med jernbane 2015K1 - 2022K3.* URL: https://www.ssb.no/statbank/table/13482/ (visited on 20/02/2023).

Shon, Hyounguk et al. (2022). 'DLCFT: Deep Linear Continual Fine-Tuning for General Incremental Learning'. In: *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII.* Springer, pp. 513–529.

Simplilearn (2023). *Introduction to Data Imputation.* URL: https://www.simplilearn.com/data-imputation-article (visited on 25/04/2023).

SJ (2023). *Dovrebanen.* URL: https://www.sj.no/content/dam/nord/Dovrebanen.pdf (visited on 19/04/2023).

Skjøren, Caroline (Dec. 2022). 'Evaluation of train delay estimates'. MA thesis. Trondheim, 7491, Norway: Norwegian University of Science and Technology.

Solinen, Emma, Gemma Nicholson and Anders Peterson (2017). 'A microscopic evaluation of railway timetable robustness and critical points'. In: *Journal of rail transport planning & management* 7.4, pp. 207–223.

Sørensen, Anette Østbø, Johannes Bjelland et al. (2018). 'Use of mobile phone data for analysis of number of train travellers'. In: *Journal of Rail Transport Planning & Management* 8.2, pp. 123–144.

Sørensen, Anette Østbø, Andreas Dypvik Landmark et al. (2017). 'Method of analysis for delay propagation in a single-track network'. In: *Journal of Rail Transport Planning & Management* 7.1-2, pp. 77–97.

Sørskår, Anders L. (Dec. 2022). 'Predicting Train Departure Delays'. MA thesis. Trondheim, 7491, Norway: Norwegian University of Science and Technology.

Spanninger, T et al. (2022). 'A review of train delay prediction approaches'. In: *Journal of Rail Transport Planning Management* 22, pp. 100–312.

Spanninger, Thomas, Alessio Trivella and Francesco Corman (2020a). 'Approaches for real-time train delay prediction'. In: *20th Swiss Transport Research Conference.* STRC. Zurich: STRC, p. 7.

— (2020b). 'Approaches for real-time train delay prediction'. In: *20th Swiss Transport Research Conference (STRC 2020)(virtual).* STRC.

Stamos, Iraklis et al. (2015). 'Impact assessment of extreme weather events on transport networks: A data-driven approach'. In: *Transportation research part D: transport and environment* 34, pp. 168–178.

Statens vegvesen (2019). *Vedlegg A - NeTEx profil Norge.* https://vegvesen.brage.unit.no/vegvesen-xmlui/bitstream/handle/11250/2583754/HB%20N801%20vedlegg.pdf?sequence=2&isAllowed=y.

Stradling, Stephen G, Jillian Anable and Michael Carreno (2007). 'Performance, importance and user disgruntlement: A six-step method for measuring satisfaction with travel modes'. In: *Transportation Research Part A: Policy and Practice* 41.1, pp. 98–106.

Sun, Lishan et al. (2018). 'Vulnerability assessment of urban rail transit based on multi-static weighted method in Beijing, China'. In: *Transportation Research Part A: Policy and Practice* 108, pp. 12–24.

Taleongpong, Panukorn et al. (2022). 'Machine learning techniques to predict reactionary delays and other associated key performance indicators on British railway network'. In: *Journal of Intelligent Transportation Systems* 26.3, pp. 311–329.

Tang, Yun and Shuping Huang (2019). 'Assessing seismic vulnerability of urban road networks by a Bayesian network approach'. In: *Transportation research part D: transport and environment* 77, pp. 390–402.

Thaduri, Adithya, Diego Galar and Uday Kumar (2015). 'Railway assets: A potential domain for big data analytics'. In: *Procedia Computer Science* 53, pp. 457–467.

Tiong, Kah Yong, Zhenliang Ma and Carl-William Palmqvist (2023a). 'A review of data-driven approaches to predict train delays'. In: *Transportation Research Part C: Emerging Technologies* 148, p. 104027.

— (2023b). 'Analyzing Factors Contributing to Real-time Train Arrival Delays using Seemingly Unrelated Regression Models'. In.

— (2022). 'Real-time Train Arrival Time Prediction at Multiple Stations and Arbitrary Times'. In: *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 793–798.

To, WM (2015). 'Centrality of an urban rail system'. In: *Urban Rail Transit* 1.4, pp. 249–256.

Too, Edna Chebet et al. (2019). 'A comparative study of fine-tuning deep learning models for plant disease identification'. In: *Computers and Electronics in Agriculture* 161, pp. 272–279.

Tukey, John W (1972). 'Exploratory data analysis as part of a larger whole'. In: *Proceedings of the 18th conference on design of experiments in Army research and development I. Washington, DC*. Vol. 1010, pp. 15–27.

Turing, Alan M. (2009). 'Computing Machinery and Intelligence'. In: *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*. Ed. by Robert Epstein, Gary Roberts and Grace Beber. Dordrecht: Springer Netherlands, pp. 23–65. ISBN: 978-1-4020-6710-5. DOI: 10.1007/978-1-4020-6710-5_3. URL: https://doi.org/10.1007/978-1-4020-6710-5_3.

Vegdirektoratet, Statens vegvesen (2019). *Nasjonale rutedata-rammer og informasjonselementer: normal [Håndbok N801]*.

Wand, Yair and Richard Y Wang (1996). 'Anchoring data quality dimensions in ontological foundations'. In: *Communications of the ACM* 39.11, pp. 86–95.

Wang, Haifeng et al. (2013). 'A topology-based model for railway train control systems'. In: *IEEE transactions on intelligent transportation systems* 14.2, pp. 819–827.

Wang, Pu and Qing-peng Zhang (2019). 'Train delay analysis and prediction based on big data fusion'. In: *Transportation Safety and Environment* 1.1, pp. 79–88.

Wen, Chao, Ping Huang, Zhongcan Li, Javad Lessan et al. (2019a). 'Train Dispatching Management With Data- Driven Approaches: A Comprehensive Review and Appraisal'. In: *IEEE Access* 7, pp. 114547–114571. DOI: 10.1109/ACCESS.2019.2935106.

— (2019b). 'Train Dispatching Management With Data-Driven Approaches: A Comprehensive Review and Appraisal'. In: *IEEE Access* 7, pp. 114547–114571. DOI: 10.1109/ACCESS.2019.2935106.

Wen, Chao, Ping Huang, Zhongcan Li and Weiwei Mou (2019). 'A predictive model of train delays on a railway line'. In: *Journal of Forecasting* 39, pp. 470–488.

Wu, Jianqing et al. (2022). 'The Bounds of Improvements Toward Real-Time Forecast of Multi-Scenario Train Delays'. In: *IEEE Transactions on Intelligent Transportation Systems* 23.3, pp. 2445–2456. DOI: 10.1109/TITS.2021.3099031.

Xia, Yuanni et al. (2013). 'Railway infrastructure disturbances and train operator performance: The role of weather'. In: *Transportation research part D: transport and environment* 18, pp. 97–102.

Xu, Donna et al. (2019). 'A data-analytics approach for enterprise resilience'. In: *IEEE Intelligent Systems* 34.3, pp. 6–18.

Xu, Yongjun et al. (2021). 'Artificial intelligence: A powerful paradigm for scientific research'. In: *The Innovation* 2.4, p. 100179.

Yaghini, Masoud, Mohammad M Khoshraftar and Masoud Seyedabadi (2013). 'Railway passenger train delay prediction via neural network model'. In: *Journal of advanced transportation* 47.3, pp. 355–368.

Yang, Li and Abdallah Shami (2020). 'On hyperparameter optimization of machine learning algorithms: Theory and practice'. In: *Neurocomputing* 415, pp. 295–316.

Yang, Xin et al. (2018). 'Recognizing the critical stations in urban rail networks: an analysis method based on the smart-card data'. In: *IEEE Intelligent Transportation Systems Magazine* 11.1, pp. 29–35.

Ye, Qian and Hyun Kim (2019). 'Assessing network vulnerability of heavy rail systems with the impact of partial node failures'. In: *Transportation* 46.5, pp. 1591–1614.

Yin, Jiateng et al. (2022). 'Quantitative analysis for resilience-based urban rail systems: A hybrid knowledge-based and data-driven approach'. In: *Reliability Engineering & System Safety* 219, p. 108183.

Yuan, J, RMP Goverde and IA Hansen (2002). 'Propagation of train delays in stations'. In: *WIT Transactions on The Built Environment* 61.

Yuan, Jianxin (2006). *Stochastic modelling of train delays and delay propagation in stations*. Vol. 2006. Eburon Uitgeverij BV.

Yuan, Jianxin and Ingo A Hansen (2007). 'Optimizing capacity utilization of stations by estimating knock-on train delays'. In: *Transportation Research Part B: Methodological* 41.2, pp. 202–217.

Yuan, Jianxin and Ingo Arne Hansen (2002). 'Punctuality of train traffic in dutch railway stations'. In: *Traffic and transportation studies: Proceedings of ICTTS Guilin*, pp. 23–25.

Zakeri, Ghazal and Nils Olsson (2018). 'Investigating the effect of weather on punctuality of Norwegian railways: a case study of the Nordland Line'. In: *Journal of Modern Transportation* 26.4, pp. 255–267.

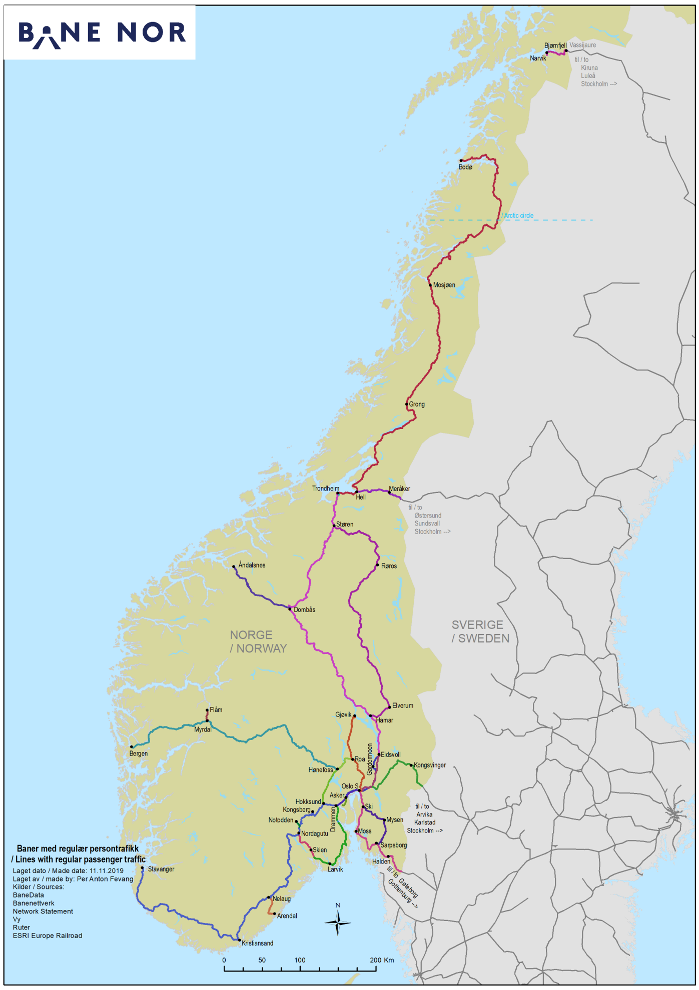
Zakeri, Ghazal and Nils OE Olsson (2018). 'Investigating the effect of weather on punctuality of Norwegian railways: a case study of the Nordland Line'. In: *Journal of Modern Transportation* 26, pp. 255–267.

Zhang, Lanhua et al. (2013). 'Modelling and optimisation on bus transport system with graph theory and complex network'. In: *International journal of computer applications in technology* 48.1, pp. 83–92.

APPENDICES

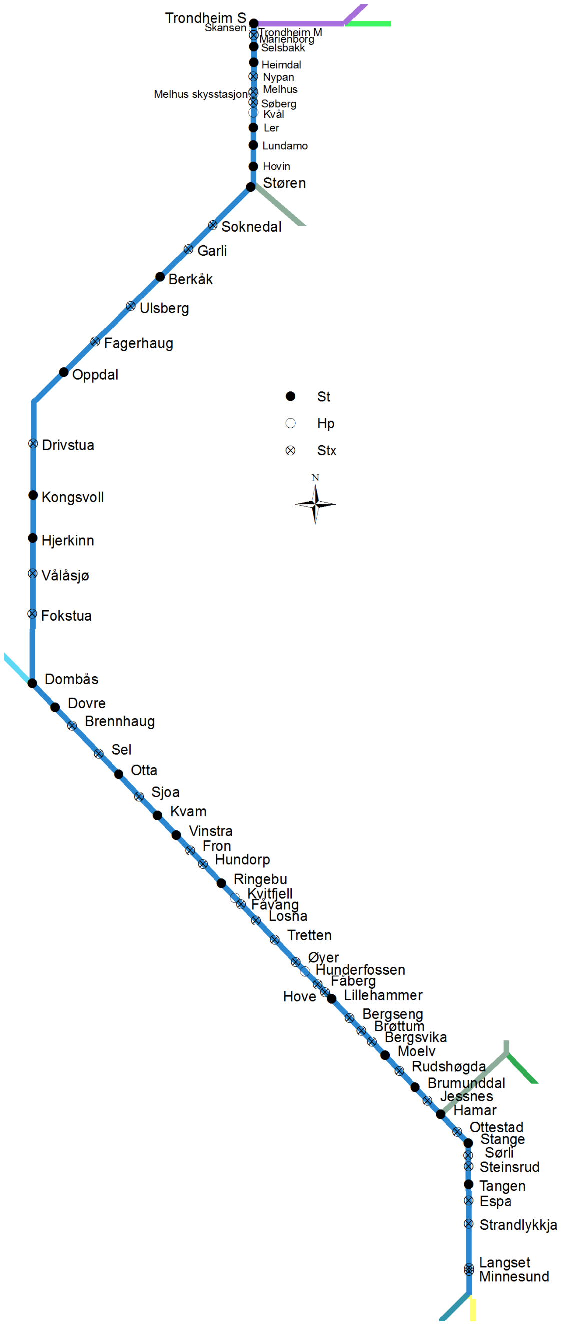
A Norwegian Railway Network

A.1 The Norwegian Railway Network Line Maps



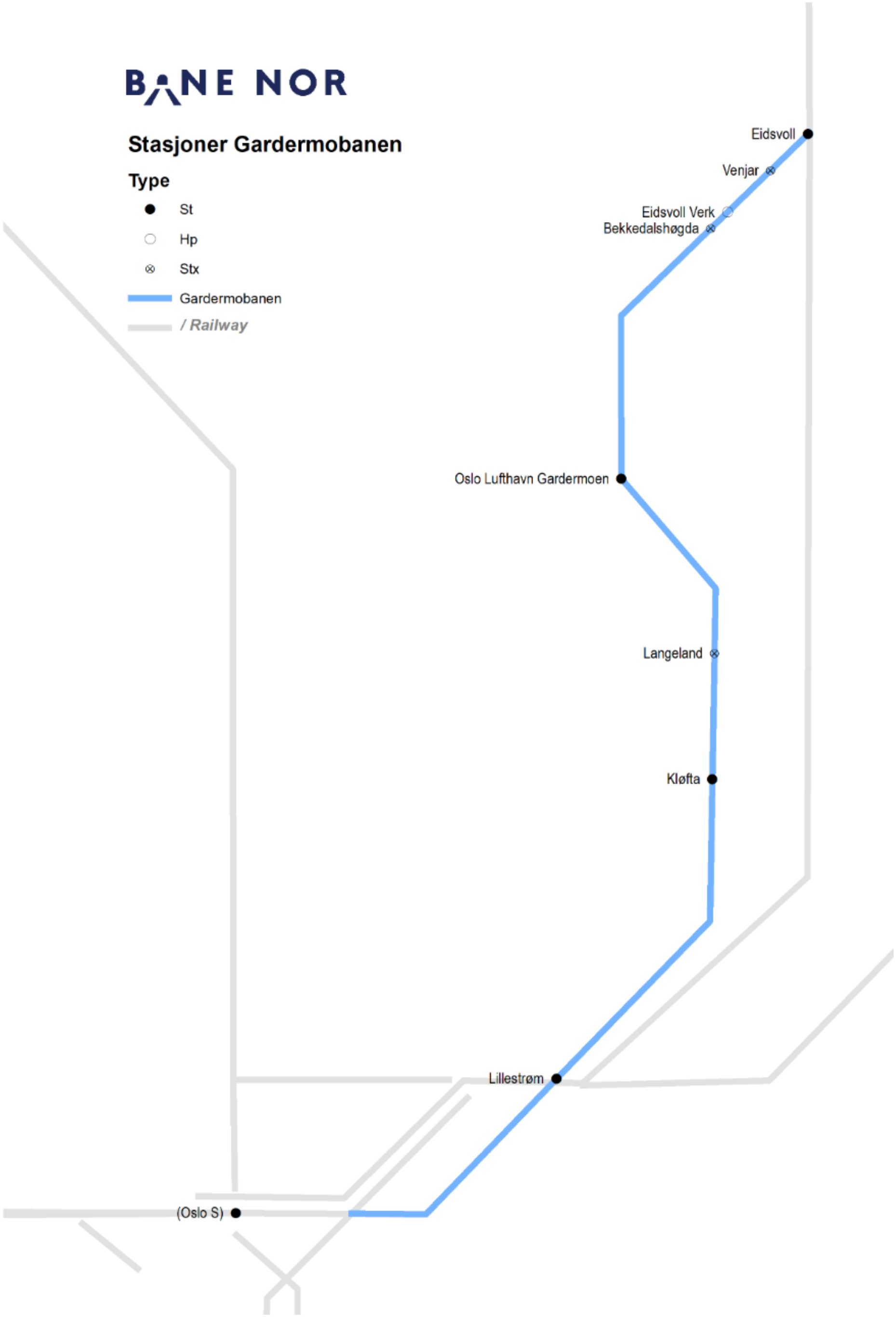
Line maps from BaneNOR (2023d)

A.2 Dovrebanen Line Maps



Line map from BaneNOR (2023c)

A.3 Gardermobanen Line Maps

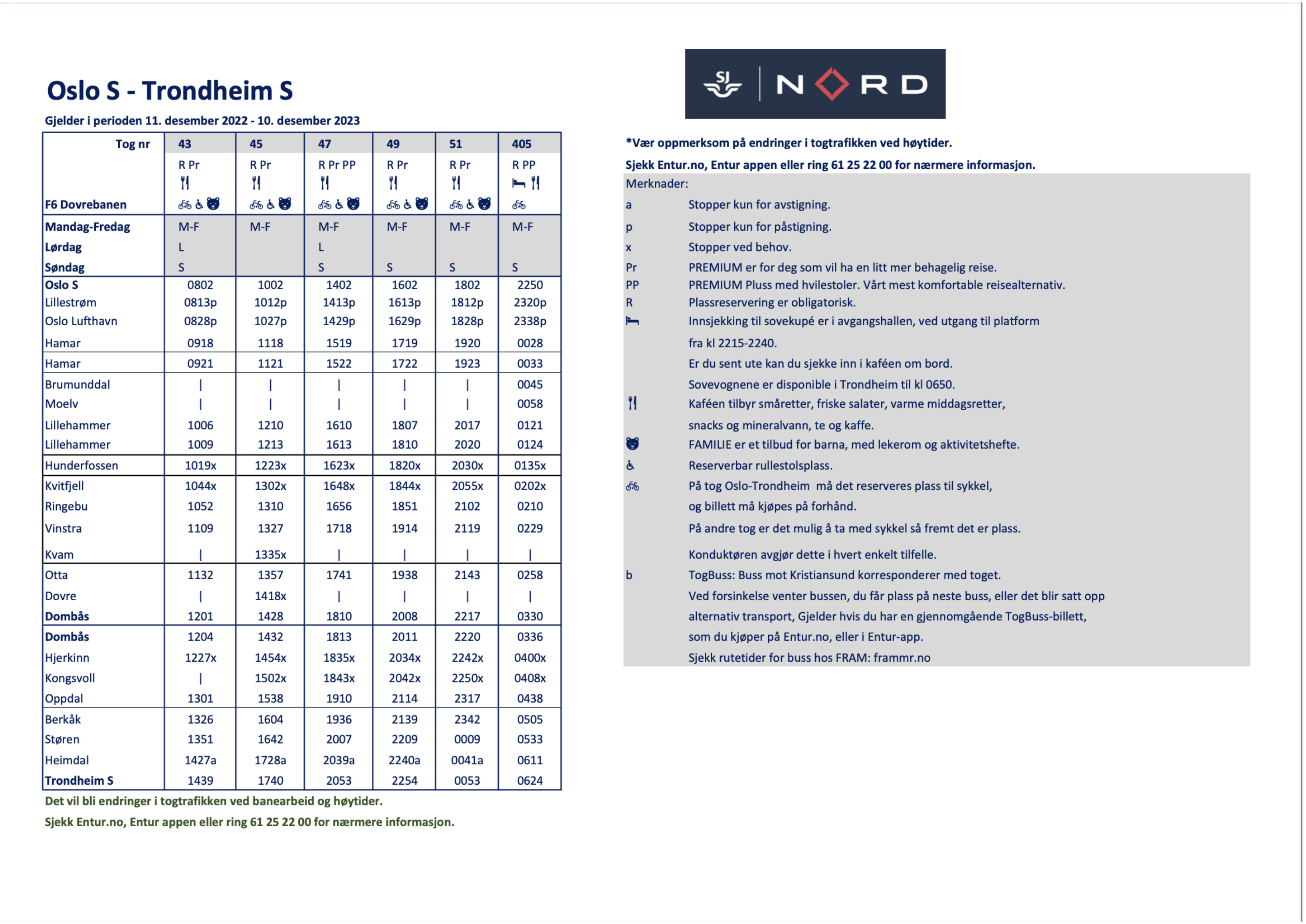


Line map from BaneNOR (2023b)

B Timetable from SJ

Routetables are gathered from SJ (2023)

B.1 Oslo - Trondheim



Oslo S - Trondheim S

Gjelder i perioden 11. desember 2022 - 10. desember 2023

Tog nr	43	45	47	49	51	405
	R Pr	R Pr	R Pr PP	R Pr	R Pr	R PP
	🍴	🍴	🍴	🍴	🍴	🛏 🍴
F6 Dovrebanen	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲
Mandag-Fredag	M-F	M-F	M-F	M-F	M-F	M-F
Lørdag	L		L			
Søndag	S		S	S	S	S
Oslo S	0802	1002	1402	1602	1802	2250
Lillestrøm	0813p	1012p	1413p	1613p	1812p	2320p
Oslo Lufthavn	0828p	1027p	1429p	1629p	1828p	2338p
Hamar	0918	1118	1519	1719	1920	0028
Hamar	0921	1121	1522	1722	1923	0033
Brumunddal	|	|	|	|	|	0045
Moelv	|	|	|	|	|	0058
Lillehammer	1006	1210	1610	1807	2017	0121
Lillehammer	1009	1213	1613	1810	2020	0124
Hunderfossen	1019x	1223x	1623x	1820x	2030x	0135x
Kvitfjell	1044x	1302x	1648x	1844x	2055x	0202x
Ringebu	1052	1310	1656	1851	2102	0210
Vinstra	1109	1327	1718	1914	2119	0229
Kvam	|	1335x	|	|	|	|
Otta	1132	1357	1741	1938	2143	0258
Dovre	|	1418x	|	|	|	|
Dombås	1201	1428	1810	2008	2217	0330
Dombås	1204	1432	1813	2011	2220	0336
Hjerkinn	1227x	1454x	1835x	2034x	2242x	0400x
Kongsvoll	|	1502x	1843x	2042x	2250x	0408x
Oppdal	1301	1538	1910	2114	2317	0438
Berkåk	1326	1604	1936	2139	2342	0505
Støren	1351	1642	2007	2209	0009	0533
Heimdal	1427a	1728a	2039a	2240a	0041a	0611
Trondheim S	1439	1740	2053	2254	0053	0624

Det vil bli endringer i togtrafikken ved banearbeid og høytider.

Sjekk Entur.no, Entur appen eller ring 61 25 22 00 for nærmere informasjon.

***Vær oppmerksom på endringer i togtrafikken ved høytider.**

Sjekk Entur.no, Entur appen eller ring 61 25 22 00 for nærmere informasjon.

Merknader:

a Stopper kun for avstigning.

p Stopper kun for påstigning.

x Stopper ved behov.

Pr PREMIUM er for deg som vil ha en litt mer behagelig reise.

PP PREMIUM Pluss med hvilestoler. Vårt mest komfortable reisealternativ.

R Plassreservering er obligatorisk.

🛏 Innsjekking til sovekupé er i avgangshallen, ved utgang til platform
fra kl 2215-2240.
Er du sent ute kan du sjekke inn i kaféen om bord.
Sovevognene er disponible i Trondheim til kl 0650.

🍴 Kaféen tilbyr småretter, friske salater, varme middagsretter,
snacks og mineralvann, te og kaffe.

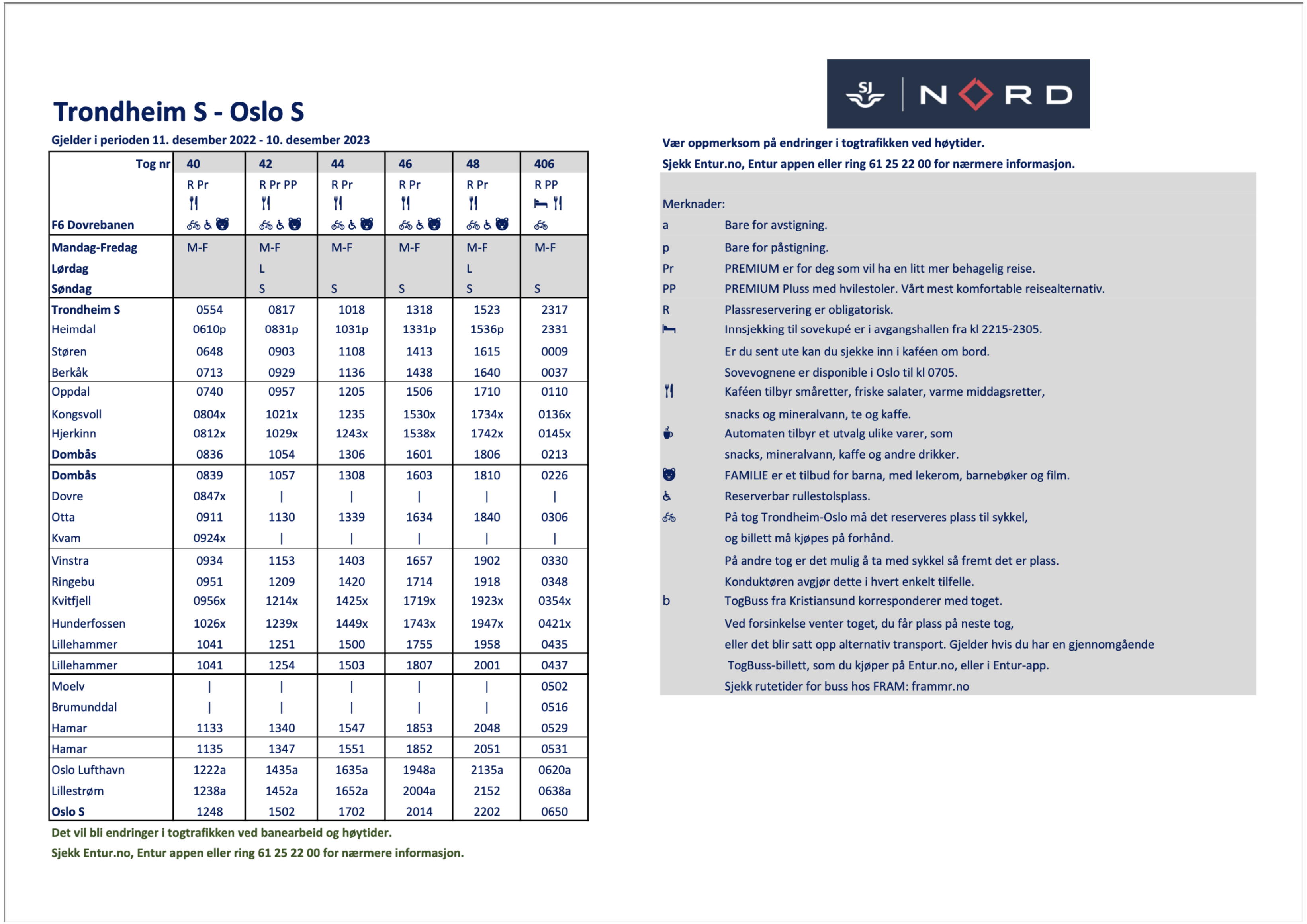
🐻 FAMILIE er et tilbud for barna, med lekerom og aktivitetshefte.

♿ Reserverbar rullestolsplass.

🚲 På tog Oslo-Trondheim må det reserveres plass til sykkel,
og billett må kjøpes på forhånd.
På andre tog er det mulig å ta med sykkel så fremt det er plass.
Konduktøren avgjør dette i hvert enkelt tilfelle.

b TogBuss: Buss mot Kristiansund korresponderer med toget.
Ved forsinkelse venter bussen, du får plass på neste buss, eller det blir satt opp
alternativ transport, Gjelder hvis du har en gjennomgående TogBuss-billett,
som du kjøper på Entur.no, eller i Entur-app.
Sjekk rutetider for buss hos FRAM: frammr.no

B.2 Trondheim - Oslo



Trondheim S - Oslo S

Gjelder i perioden 11. desember 2022 - 10. desember 2023

Tog nr	40	42	44	46	48	406
	R Pr	R Pr PP	R Pr	R Pr	R Pr	R PP
	🍴	🍴	🍴	🍴	🍴	🛏 🍴
F6 Dovrebanen	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲 ♿ 🐻	🚲
Mandag-Fredag	M-F	M-F	M-F	M-F	M-F	M-F
Lørdag		L			L	
Søndag		S	S	S	S	S
Trondheim S	0554	0817	1018	1318	1523	2317
Heimdal	0610p	0831p	1031p	1331p	1536p	2331
Støren	0648	0903	1108	1413	1615	0009
Berkåk	0713	0929	1136	1438	1640	0037
Oppdal	0740	0957	1205	1506	1710	0110
Kongsvoll	0804x	1021x	1235	1530x	1734x	0136x
Hjerkinn	0812x	1029x	1243x	1538x	1742x	0145x
Dombås	0836	1054	1306	1601	1806	0213
Dombås	0839	1057	1308	1603	1810	0226
Dovre	0847x	|	|	|	|	|
Otta	0911	1130	1339	1634	1840	0306
Kvam	0924x	|	|	|	|	|
Vinstra	0934	1153	1403	1657	1902	0330
Ringebu	0951	1209	1420	1714	1918	0348
Kvitfjell	0956x	1214x	1425x	1719x	1923x	0354x
Hunderfossen	1026x	1239x	1449x	1743x	1947x	0421x
Lillehammer	1041	1251	1500	1755	1958	0435
Lillehammer	1041	1254	1503	1807	2001	0437
Moelv	|	|	|	|	|	0502
Brumunddal	|	|	|	|	|	0516
Hamar	1133	1340	1547	1853	2048	0529
Hamar	1135	1347	1551	1852	2051	0531
Oslo Lufthavn	1222a	1435a	1635a	1948a	2135a	0620a
Lillestrøm	1238a	1452a	1652a	2004a	2152	0638a
Oslo S	1248	1502	1702	2014	2202	0650

Det vil bli endringer i togtrafikken ved banearbeid og høytider.

Sjekk Entur.no, Entur appen eller ring 61 25 22 00 for nærmere informasjon.

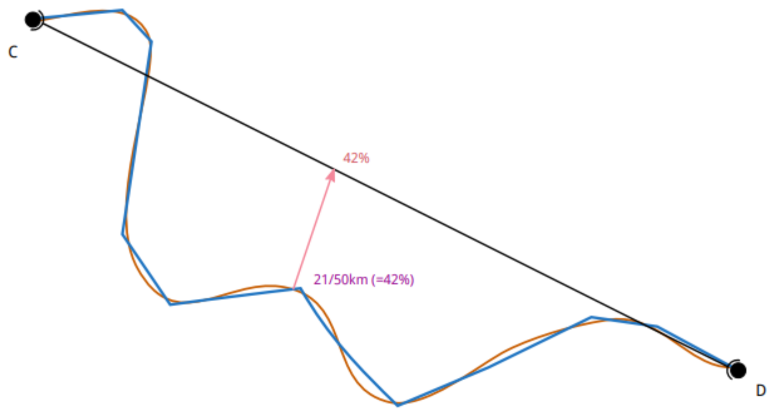
Vær oppmerksom på endringer i togtrafikken ved høytider.

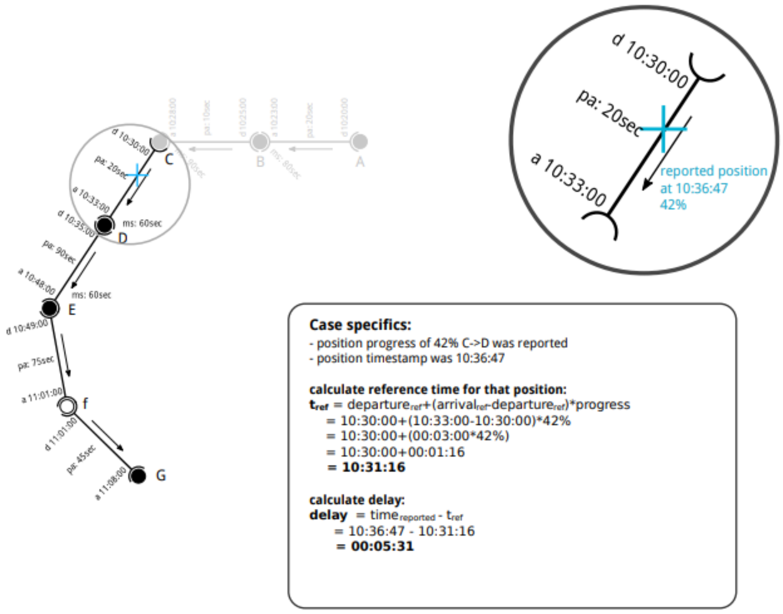
Sjekk Entur.no, Entur appen eller ring 61 25 22 00 for nærmere informasjon.

Merknader:

a	Bare for avstigning.
p	Bare for påstigning.
Pr	PREMIUM er for deg som vil ha en litt mer behagelig reise.
PP	PREMIUM Pluss med hvilestoler. Vårt mest komfortable reisealternativ.
R	Plassreservering er obligatorisk.
🛏	Innsjekking til sovekupé er i avgangshallen fra kl 2215-2305. Er du sent ute kan du sjekke inn i kaféen om bord. Sovevognene er disponible i Oslo til kl 0705.
🍴	Kaféen tilbyr småretter, friske salater, varme middagsretter, snacks og mineralvann, te og kaffe.
☕	Automaten tilbyr et utvalg ulike varer, som snacks, mineralvann, kaffe og andre drikker.
🐻	FAMILIE er et tilbud for barna, med lekerom, barnebøker og film.
♿	Reserverbar rullestolsplass.
🚲	På tog Trondheim-Oslo må det reserveres plass til sykkel, og billett må kjøpes på forhånd. På andre tog er det mulig å ta med sykkel så fremt det er plass. Konduktøren avgjør dette i hvert enkelt tilfelle.
b	TogBuss fra Kristiansund korresponderer med toget. Ved forsinkelse venter toget, du får plass på neste tog, eller det blir satt opp alternativ transport. Gjelder hvis du har en gjennomgående TogBuss-billett, som du kjøper på Entur.no, eller i Entur-app. Sjekk rutetider for buss hos FRAM: frammr.no

C Figures of Bane NOR´s Train Prediction Model





Case specifics:

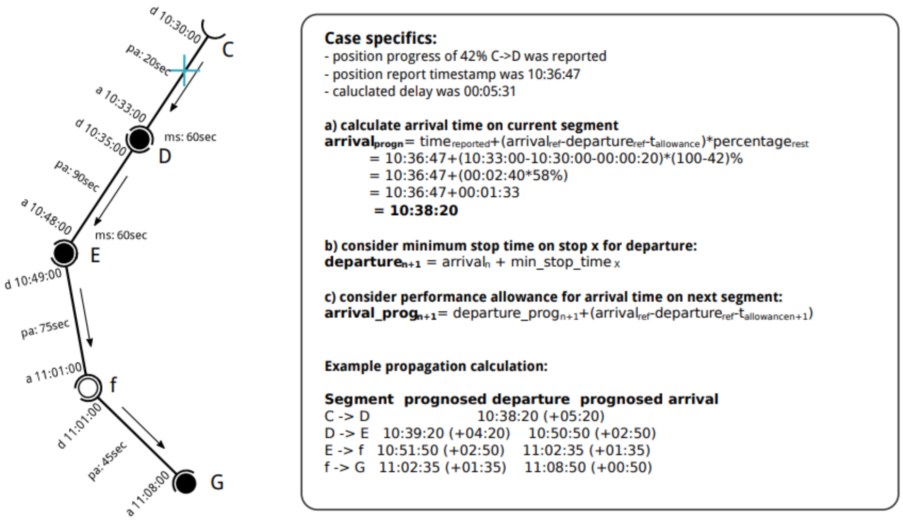
- position progress of 42% C->D was reported
- position timestamp was 10:36:47

calculate reference time for that position:

t_{ref} = departure$_{ref}$+(arrival$_{ref}$-departure$_{ref}$)*progress
= 10:30:00+(10:33:00-10:30:00)*42%
= 10:30:00+(00:03:00*42%)
= 10:30:00+00:01:16
= 10:31:16

calculate delay:

delay = time$_{reported}$ - t_{ref}
= 10:36:47 - 10:31:16
= 00:05:31



Case specifics:

- position progress of 42% C->D was reported
- position report timestamp was 10:36:47
- caluclated delay was 00:05:31

a) calculate arrival time on current segment

arrival$_{\mathbf{progn}}$= time$_{reported}$+(arrival$_{ref}$-departure$_{ref}$-t$_{allowance}$)*percentage$_{rest}$
= 10:36:47+(10:33:00-10:30:00-00:00:20)*(100-42)%
= 10:36:47+(00:02:40*58%)
= 10:36:47+00:01:33
= 10:38:20

b) consider minimum stop time on stop x for departure:

departure$_{\mathbf{n+1}}$ = arrival$_n$ + min_stop_time$_x$

c) consider performance allowance for arrival time on next segment:

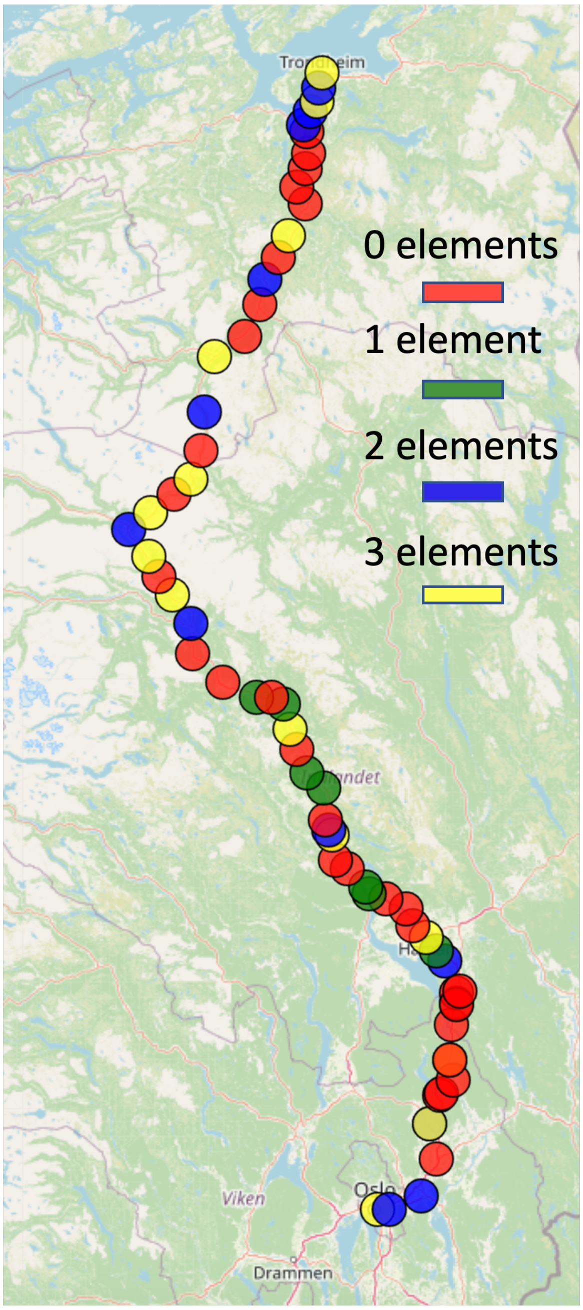
arrival_prog$_{\mathbf{n+1}}$= departure_prog$_{n+1}$+(arrival$_{ref}$-departure$_{ref}$-t$_{allowancen+1}$)

Example propagation calculation:

Segment	**prognosed departure**	**prognosed arrival**
C -> D		10:38:20 (+05:20)
D -> E	10:39:20 (+04:20)	10:50:50 (+02:50)
E -> f	10:51:50 (+02:50)	11:02:35 (+01:35)
f -> G	11:02:35 (+01:35)	11:08:50 (+00:50)

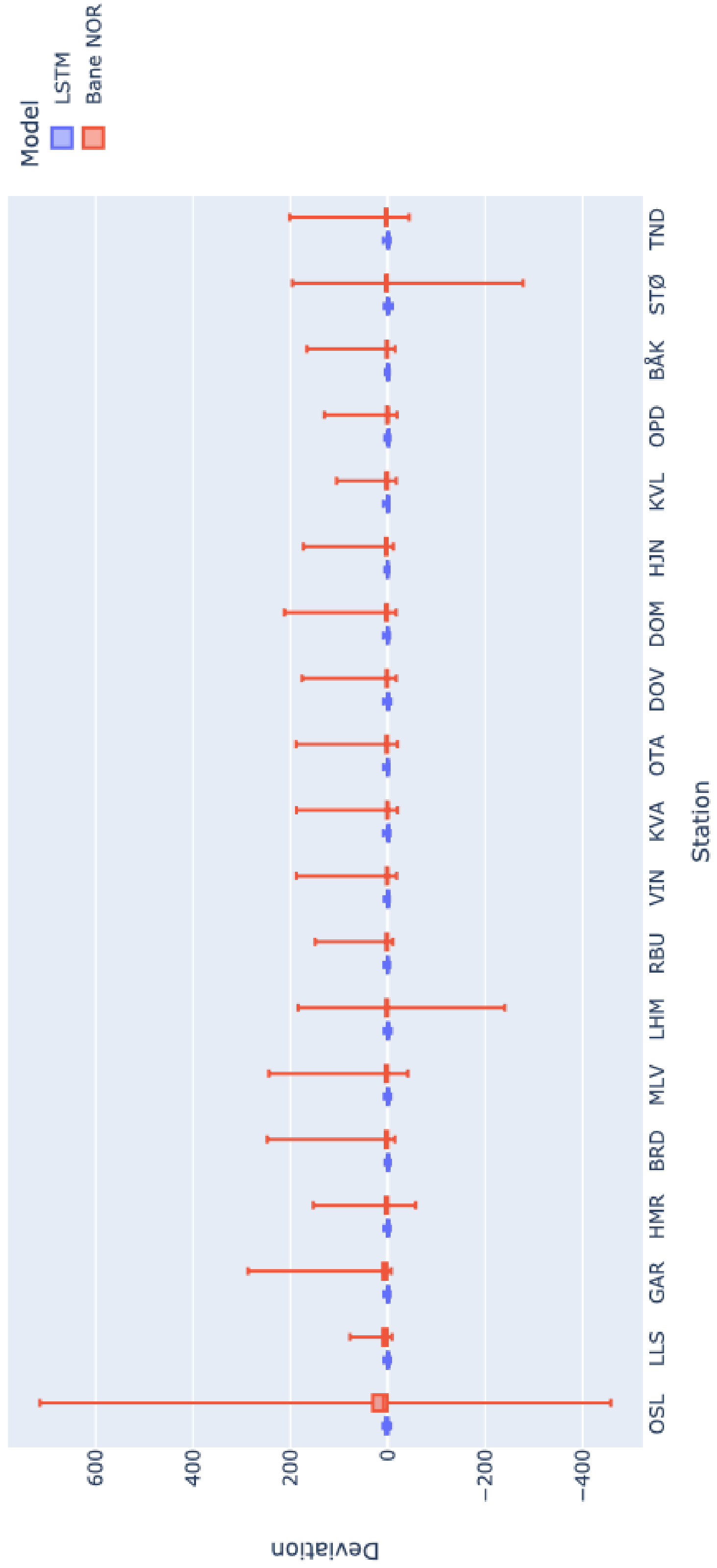
D Data availability along the railway line

Data availability along the railway line. The number of elements refer to how many weather elements were available at the given weather station.

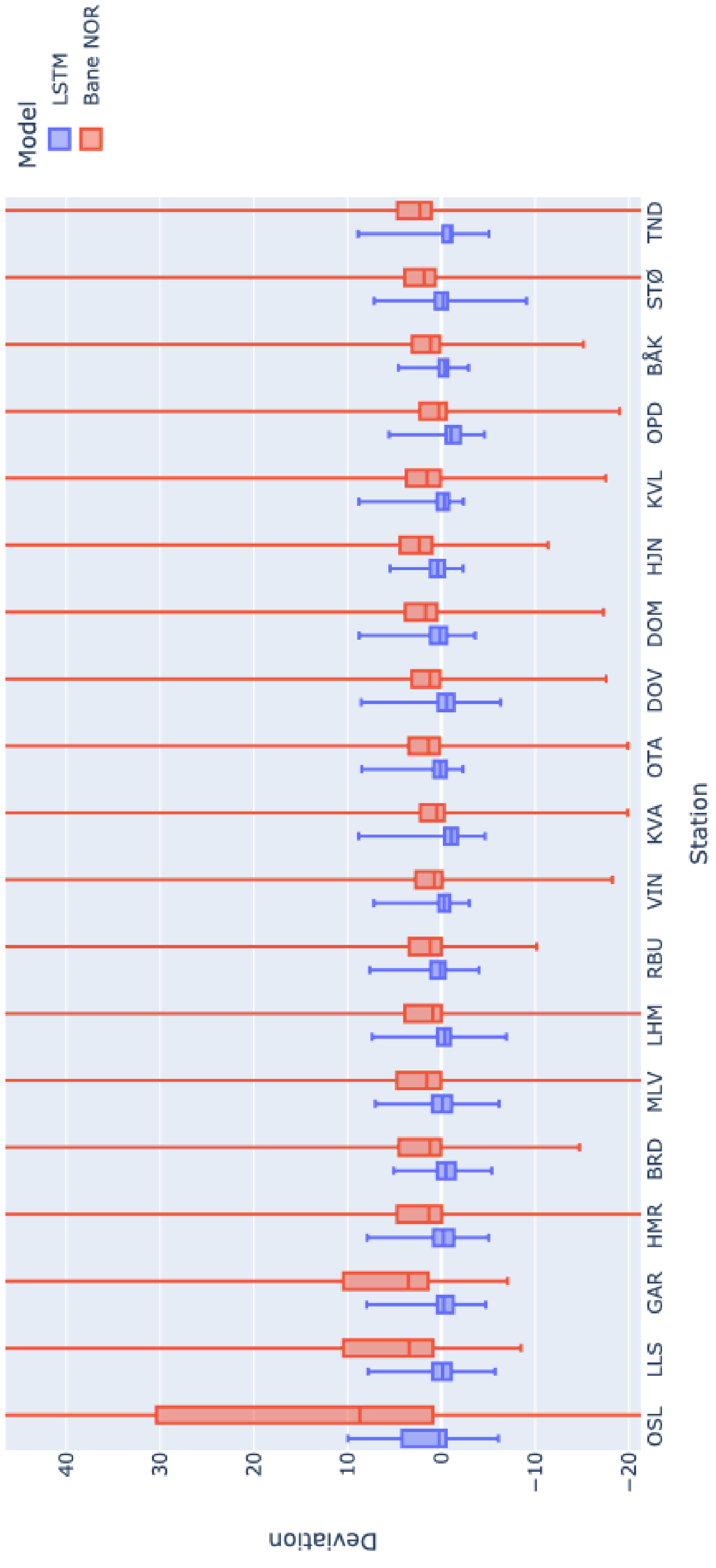


E Results: Visual Inspection

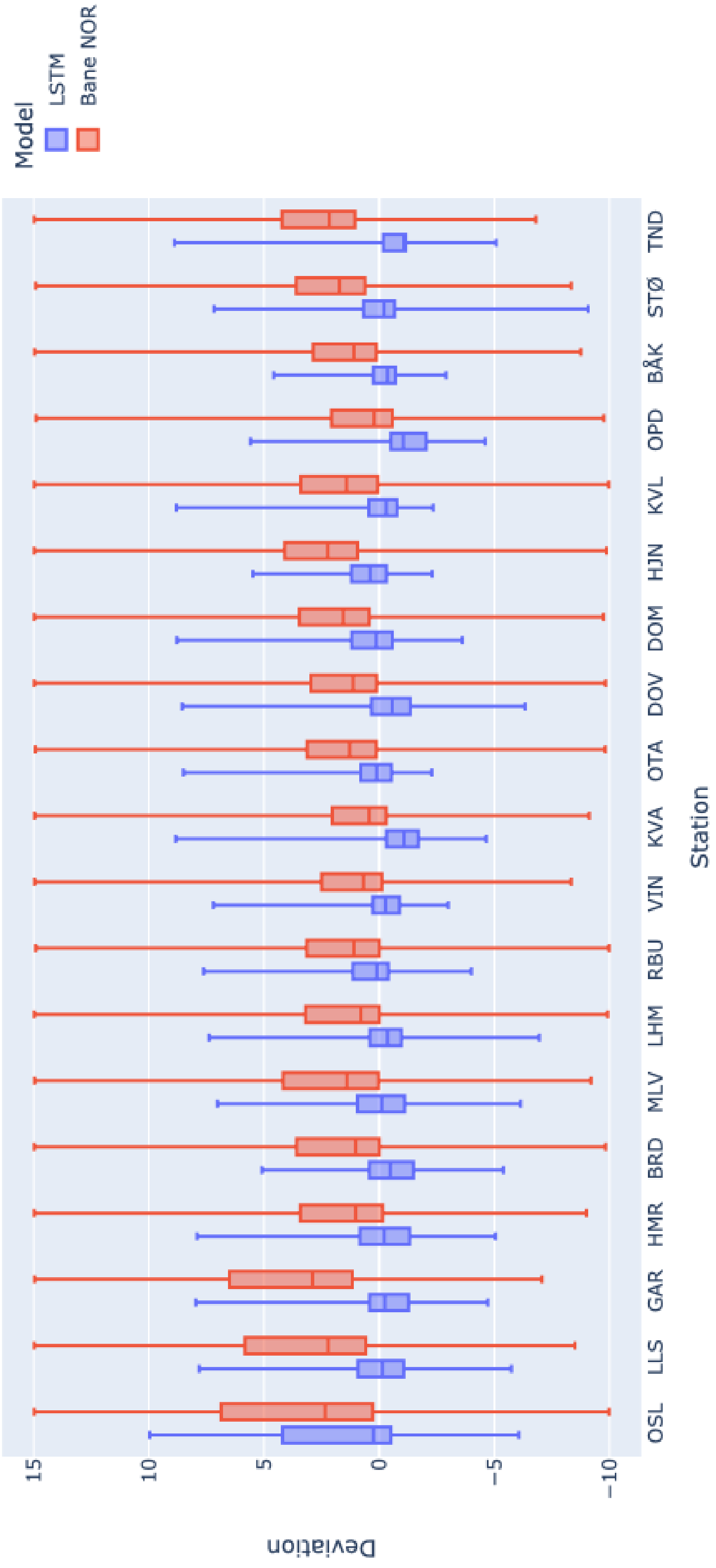
E.1 Box plot from figure 6.3.1



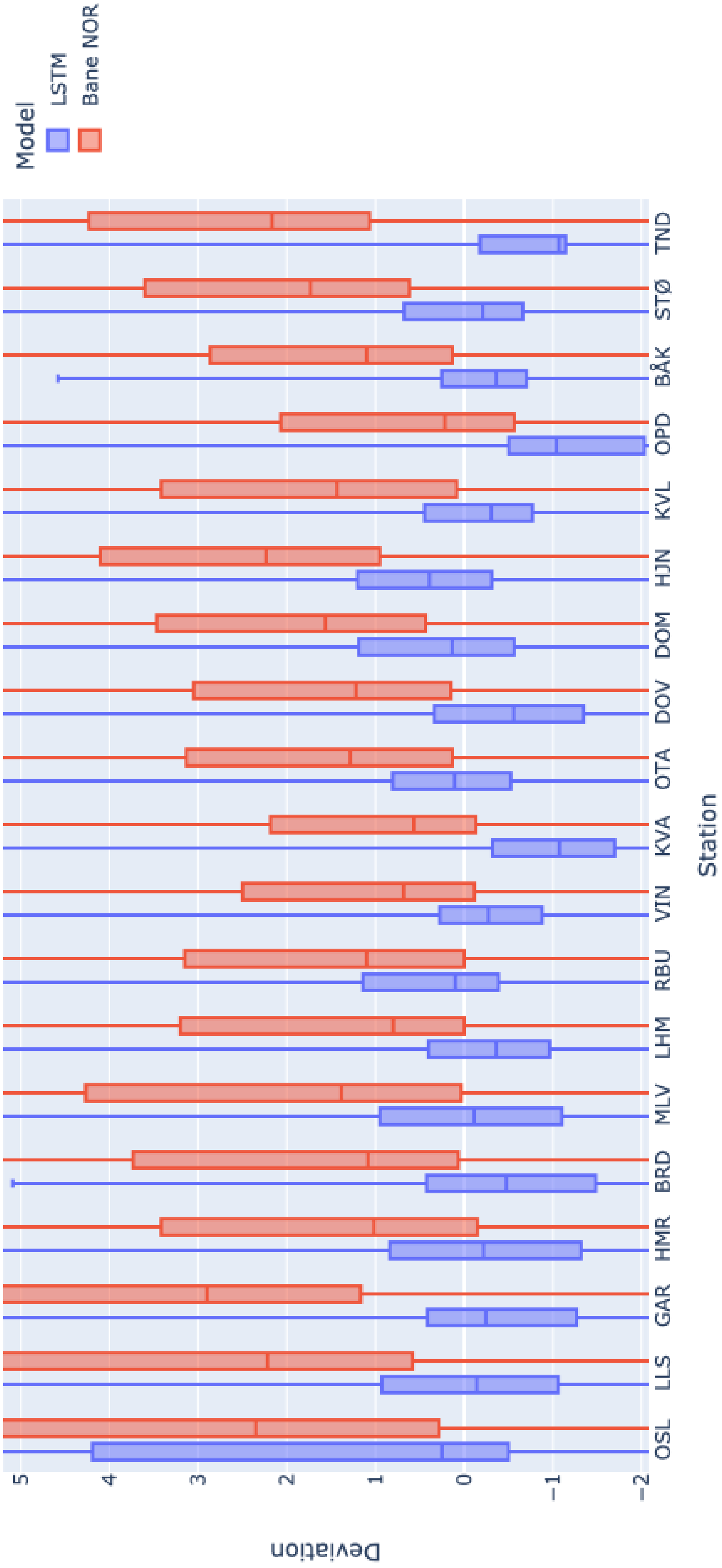
E.2 Box plot from figure 6.3.2



E.3 Box plot from figure 6.3.3



E.4 Box plot from figure 6.3.4



F Long Short-Term Memory (LSTM)

F.1 Parameters and Functions

Dropout Probability: Dropout is a regularization technique used to mitigate the risk of overfitting in neural networks. It randomly sets a fraction of the input units to 0 at each training step. This helps to reduce interdependent learning among the neurons. The dropout probability dictates the fraction of input units to be set to zero out during training (Goodfellow et al. 2016).

Learning Rate: The learning rate is a hyper-parameter that governs the step size at which the model parameters are updated during training. A higher learning rate allows the model to learn faster but may result in overshooting the optimal solution, while a lower learning rate may lead to slower convergence. The learning rate is typically set between 0 and 1 (Goodfellow et al. 2016).

Weight Decay: Weight decay, or L2 regularization, serves as a valuable technique to counteract overfitting in machine learning models. It involves augmenting the loss function with a penalty term based on the magnitude of the model weights. By doing so, weight decay promotes the learning of smaller weight values, effectively curbing the complexity of the learned function. The weight decay parameter governs the intensity of the regularization term, influencing the extent to which weight values are encouraged to be minimized during training (Goodfellow et al. 2016).

Optimiser: The optimizer serves as a critical algorithm employed to adjust the model parameters using computed gradients during the training process. It plays a pivotal role in determining how the model learns from the provided training data. Various commonly used optimisers include Stochastic Gradient Descent (SGD), Adam, RMSprop, and Adagrad. Through optimizer hyper-parameters, one can precisely control the behavior of the optimization algorithm, influencing factors such as the learning rate schedule and momentum. These hyper-parameters play a vital role in optimizing the training process and enhancing the model's performance (Goodfellow et al. (2016); PyTorch (2023)).

Activation Functions

Sigmoid Function (σ): The sigmoid function is used in the input, forget, and output gates. It is used to squash the input values into the range [0,1]. Particularly useful for deciding whether to forget, to update, or to output the cell state.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Hyperbolic Tangent Function (Tanh): The tanh function is used for creating the new cell state candidate. It is used to squash the input values into the range [-1,1], which can handle both positive and negative correlations in the data.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

F.2 Training Procedure

The training of an LSTM model involves the use of Backpropagation Through Time (BPTT) and an optimization algorithm like Stochastic Gradient Descent (SGD) or its variants. This project has employed the Adam optimiser, which is an extended variant of the SGD. However, for the sake of simplicity, the training procedure will be explained using the SGD.

We assume the input at time t to be x_t and the output to be y_t. The hidden state is represented by h_t and the cell state by c_t. The model parameters include the weight matrices W, U, and bias vectors b. Subscripts i, f, o, and c denote the input gate, forget gate, output gate, and cell state, respectively.

Forward Pass

The LSTM cell computations for a given time step t are as follows:

- Input gate: $i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$
- Forget gate: $f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$
- Output gate: $o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$
- New memory cell: $\tilde{c_t} = \tanh(W_c x_t + U_c h_{t-1} + b_c)$
- Final cell state: $c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c_t}$
- Hidden state: $h_t = o_t \circ \tanh(c_t)$

Here, $\circ$ denotes the Hadamard (element-wise) product and σ represents the sigmoid function. (PyTorch 2023)

Compute Loss

The loss for a single time step is computed as the mean squared error between the predicted output ($\hat{y_t}$) and the actual output (y_t). The total loss is the sum of the individual losses across all time steps.

$$L = \frac{1}{T} \sum_t (\hat{y_t} - y_t)^2$$

Backward Pass

Gradients of the loss function with respect to the model parameters are computed using the chain rule of differentiation. For instance, the gradient of L_t with respect to the input gate weights W_{ii} is computed as follows:

$$\frac{\partial L_t}{\partial W_{ii}} = \frac{\partial L_t}{\partial \hat{y_t}} \frac{\partial \hat{y_t}}{\partial h_t} \frac{\partial h_t}{\partial i_t} \frac{\partial i_t}{\partial W_{ii}}$$

These gradients are then used to update the model parameters.(PyTorch 2023)

Update Parameters

Model parameters are updated in the direction that minimizes the loss. For instance, the update for W_{ii} using SGD with learning rate η is as follows:

$$W_{ii} = W_{ii} - \eta \frac{\partial L}{\partial W_{ii}}$$

The same procedure is followed to update all other parameters. This process is repeated for a number of iterations, or *epochs*, or until the model's performance on a validation set stops improving. (Goodfellow et al. 2016)

NTNU

Norwegian University of
Science and Technology