FACULTY OF SCIENCE AND TECHNOLOGY

MASTER'S THESIS

Study programme / specialisation:
MASTER IN DATA SCIENCE

The *(spring/autumn)* semester, *(year)*
SPRING, 2023

[✓ Open] / Confidential

Author:
AMAN RIAZ

Supervisor at UiS: ASSOCIATE PROFESSOR NAEEM KHADEMI

Co-supervisor: ASSOCIATE PROFESSOR ERLEND TØSSEBRO

External supervisor(s):

Thesis title:
TUNNEL TRAFFIC FORECASTING USING DEEP LEARNING

Credits (ECTS): 30

Keywords:
Deep learning, tunnel safety
traffic forecasting, time-series
prediction, neural networks

Pages: 59
+ appendix:

Stavanger, *(date)* 15.06.2023

Master in Data Science

Tunnel Traffic Forecasting Using Deep Learning

Universitetet
i Stavanger

Faculty of Science & Technology

Aman Riaz

Spring, 2023

15.06.2023

Abstract

Tunnel traffic congestion can increase the risk of traffic accidents, tunnel fires, and environmental effect. Despite numerous studies on traffic forecasting using deep learning, research on tunnel traffic remains limited.Utilizing traffic flow data from the Norwegian Public Road Administration, this thesis analyzes the applicability of recurrent neural networks for tunnel traffic prediction. The data is retrieved from different sources and traffic sensors near or inside the tunnels are selected through a geo-spatial analysis. The recurrent neural network is designed to be trained on either a single tunnel or several tunnels. Furthermore, based on their geographical location and population density, the tunnels are classified as urban or sub-urban. Based on the results of the experiments and the sample of tunnels used, the recurrent neural network outperformed the baseline for urban tunnels in terms of root-mean-squared-error. However, the performance advantage was not significant for sub-urban tunnels. The addition of features such as temporal features and category features provided no significant results. These findings are discussed in the final sections of the thesis.

Contents

Acknowledgements

I would like to thank my two supervisors, Associate Professor Naeem Khademi & Associate Professor Erlend Tøssebro for their valuable guidance, support, and expertise throughout the process of completing this work

Also a big thank you to Jan Kristian Jensen from Statens Vegvesen for his guidance on NVDB

Finally, thank you to my family and husband for their everlasting support. And to my little one, Momina

Chapter 1

Introduction

The rapid growth of intelligent systems and the increasing production of data in recent decades have given rise to the emergence of Intelligent Transport Systems (ITS). ITS offers significant potential to revolutionize traffic management and mitigate the negative effects of traffic congestion in tunnels through the deployment of effective strategies [1]. The negative effects include substantial time delays, a higher risk of traffic accidents, and fuel wastage. By harnessing the power of the available traffic data by NPRA, ITS can potentially enhance traffic safety and lead to a better understanding of tunnel traffic dynamics.

1.1 Tunnels in Norway

Norway is a country distinguished by its mountainous landscape and a huge network of over 1260 tunnels (2022) [2]. The tunnels represent a critical component of transportation infrastructure, particularly in mountainous regions, and ensuring their safety is essential for public welfare. Norway hosts a diverse range of tunnels, including both road and underwater tunnels. One example is the Lærdal tunnel, spanning 24,5 km on the road and is the longest road tunnel in the world [2].

1.2 Traffic accidents in tunnels

According to the Fire and Explosion Protection Act (2002), tunnels exceeding a length of 500 meters are designated as separate fire objects, necessitating the implementation of supervision and emergency plans [3]. Recent technological advancements have facilitated the development of advanced vehicles equipped with features such as "lanekeeping" and "blind-zone alerts" [3]. Although tunnel accidents may occur less frequently than on open roads, they often result in more severe consequences. These incidents predominantly occur near tunnel entrances, often instigated by factors such as tire punctures or empty fuel tanks [3]. Collisions and other types of accidents account for approximately 15% of these occurrences [3]. Furthermore, Norway has experienced a significant number of tunnel fires, particularly in tunnels with steep gradients. Longer and steeper tunnels inherently carry a greater risk of fires originating from the engines or brakes of heavy vehicles. For instance, in 2019, the Gudvanga tunnel witnessed a fire caused by a heavy vehicle, leading to injuries and the tunnel's subsequent closure for a time [4]. Tunnel fires also subject surrounding rock structures to intense heat, potentially leading to structural cracks when exposed to cold mountain conditions. Therefore, comprehensive safety measures and thorough inspections are necessary before reopening a tunnel. Nonetheless, it is crucial to recognize that these precautions can disrupt commuters who rely on the tunnel for travel to work or school, and emergency response agencies.

1.3 Early detection of risk factors

Prevention and early detection of risk factors within tunnels can play a crucial role in ensuring tunnel safety. Among these risk factors, traffic congestion emerges as a significant concern. To address these risks, monitoring systems leveraging advanced technologies such as sensors, cameras, and machine learning algorithms are used. Such systems can proactively detect potential hazards and enable tunnel operators to take preventive measures. Machine learning models can utilize historic traffic data from tunnels, to predict the traffic levels h steps into the future. These predictions, in turn, enable first responders to proactively manage the traffic situation and implement suitable mitigation strategies accordingly.

1.4 Available data for tunnel traffic

There is a limitation of open source data sets about tunnel traffic, that can be utilized for the purpose of this thesis. One of the reasons for this could be that traffic in tunnels require some necessary preparations such as filtering out only tunnel traffic from the rest. The Norwegian Public Road Administration owns different data sets for traffic in general and hold potential that may be utilized for tunnel traffic forecasting.

In this study, two open-source data sources from the Norwegian Public Road Administration, namely NVDB and trafikkdata.no, are utilized to compile an integrated data set of hourly traffic data for tunnels. This data set encompasses details such as tunnel names, lengths, opening years, and various other technical and informative features. Additionally, it incorporates the hourly aggregated tunnel traffic volume for each tunnel, which is derived from the corresponding nearby traffic registration point. The data analysis and training of the model is performed on the acquired data set.

1.5 Deep learning for traffic forecasting

Machine learning has emerged as a promising approach for traffic management, offering significant contributions in optimizing traffic flow and travel time, and improving existing systems.In particular, neural networks, a group of machine learning models, have displayed a promising aptitude to comprehend complex patterns and trends within traffic data and leverage these findings for forecasting future traffic patterns. Deep learning methodologies, such as neural networks, have demonstrated the ability to discern and model the potentially non-linear dynamics and spatio-temporal dependencies present in traffic data. Such complexities prove challenging for classical statistical models [1].

1.6 Problem statement

The direction of the research is guided by a literature review conducted at the outset of this study. The literature review reveals that several deep learning models have been developed that are performing well on traffic forecasting tasks. That includes long-short-term-memory, recurrent neural networks, and auto-encoders to mention some [5]. However, there is limited research on using deep learning to predict traffic flow in tunnels.

The literature review reveals that a model is not always able to maintain its efficiently when applied to data sets from different scenarios. That is because the model is not able to generalize well enough during its training, or the data provided to the model is not representative enough. This work investigates the hypothesis of whether tunnel traffic possess less complexity as regular road traffic, or if it can be modeled with a simper neural network than what has been implemented for regular road traffic. However, it is worth highlighting that the available data is hourly, meaning the granularity of the data does not reveal real-time behaviour of tunnel traffic.

1.7 Research objective

The main research objective of this paper is to investigate the usability of the currently available traffic data provided by NPRA for tunnel traffic forecasting capabilities. The forecasting task is performed using a simple recurrent neural network (RNN). Through this work the aim is to achieve a better understanding of the possible underlying patterns that can assist deep learning models to perform better.

1.8 Research questions

The following research questions are addressed in this work:

1. Q1: What is the nature in terms of trend and seasonality of tunnel

traffic data?

2. Q2: How does an RNN model trained on a single tunnel perform versus on multiple tunnels?
3. Q3: Does adding the category (urban, or sub-urban) feature or the temporal feature improve the model performance?

Chapter 2

Literature review

This section presents a literature review and key findings that provide an overview of the current state of the art and the main challenges in the field of traffic forecasting. Since there is limited research specifically focused on tunnel traffic forecasting, this review incorporates relevant studies on traffic forecasting for regular road traffic. The review emphasizes the variations in research methodologies and the diverse range of machine-learning strategies that have been employed in the field.

The literature review is presented in two parts. First, the key findings are presented in terms of the currently used model and techniques that are used to achieve improved results. The second summarizes the key challenges that are faced in this field of study,

2.1 Shift towards data-driven models

In general, advancement in traffic prediction is gravitating towards the development of more complex and sophisticated models that are capable of handling big data, dynamic input matrices and non-stationary data. Early methodologies for estimating traffic behaviour heavily relied on statistical models such as ARIMA [6]. The limitation of statistical models in capturing the inherent randomness and variability of traffic patterns has led to a shift towards data-driven models due to statistical models not able to model non-linearity.

However, the advancement of technology and the emergence of new statistical methods soon underscored the limitations of these models, particularly their inability to forecast multiple variables [6]. The data-driven models are able to capture the dynamic nature of traffic. They are able to uncover hidden patterns and find correlations and dependencies over time that can give us a better understanding of the nature of the data.

2.2 RNN for traffic forecasting

The two commonly used models for traffic prediction that have shown promising results are Long-Short-term Memory (LSTM) and Recurrent Neural Networks (RNNs) [1], [5]. Recurrent Neural Networks (RNNs) and Long Short-term Memory (LSTM) networks have proven to be successful in predicting traffic flow, specifically to capture temporal dependencies in traffic data, and have been widely used in traffic prediction studies [1]. According to [1] the LSTM and SAE models showed great results compared to the other models. In another paper [5] which predicts traffic flow with an RNN model, and auto-encoders using DataInn/AutoPass data from NPRA, the RNN and auto-encoders outperformed the other models.

2.3 Introduction of additional data features

The addition of new features into the deep learning models is an approach that has been explored in numerous studies. Moreover, researchers have been concentrating on leveraging data from different sources, including GPS and social media information, with the objective of enhancing prediction accuracy focused on traffic in work zones with both long-term and short-term forecasting [7]. Together with traffic flow at one station, they used an upstream and downstream station to evaluate performance gain[7]. Additionally, they added other features such as the workday, hour in day, and speed limit. The results however showed that these extra parameters had little significance for the models' accuracy [7].

The work in [8] looked at the relevance of the spatial and temporal features of traffic registration stations. Their results revealed that using the spatio-temporal similarities of the stations provided a higher model accuracy rather than the physically closest station [8]. As stated in both [9] and [10] the spatial and temporal features are seemed to be the most influential on the traffic. Additionally, it is suggested in [9] that the amount of data is also an important factor improving model accuracy.

[9] tested out Logistic Regression, Neural Networks and classification trees for traffic prediction using big data. The results demonstrated that an increasing window size gave an improved model accuracy. In addition they performed clustering techniques to group similar stations together, which further enhanced the accuracy levels [9].

The work presented in [11] studies the effect of normal and abnormal traffic conditions, where abnormal conditions are defined at as conditions like unforeseen incidents that have interrupted the regular flow. The paper models and compares three machine learning models using error feedback mechanism to see if has an effect. Among the used tools, KNN-based prediction models with error feedback performed well for short-term traffic predictions [11].

2.4 Key challenges identified

This section highlights the key challenges that have been identified during the literature review.

1. Urban vs inter-city highways
2. Introduction of features
3. Parameter sensitivity

2.4.1 1. Urban vs inter-city highways

One of the challenges is the lack of similar traffic scenarios across the various research efforts on traffic prediction and deep learning. This has led to a wide range of models that are fit to the data from a specific traffic setting. The literature review indicates that while no single model is universally optimal, LSTM and RNNs have over the years demonstrated the best performance in capturing different types of traffic environments [5].

Many state-of-the-art models exhibit strong performance when trained and tested on specific datasets that are representative of a particular country, region, or local traffic patterns [5]. However, this raises concerns about their ability to generalize across diverse contexts, indicating potential limitations in their design or a lack of consideration for important features [5].

To address these concerns, it is crucial to evaluate models using a wide range of traffic data from various settings. A study conducted by researchers in [1] aimed to enhance the ability to generalize their model by employing a strategic approach to data arrangement. They used three weeks of data from each month as training set, while the rest of each month was utilized as test set.

2.4.2 2. Introduction of features

The study by [5] observed that incorporating additional features such as timestamps and vehicle gaps into the feature vector negatively impacted the performance of deep neural networks. Conversely, it positively affected the performance of SSAE and RNN models. However, as per the insights from [1], feature engineering might not play a central role in the success of deep learning models. The justification for this is that deep learning models are inherently capable of discerning long-term relationships, which are predominantly present in the raw data [1].

Various attempts have been made in order to find hidden correlations between traffic and the surrounding elements. In research conducted by [12] they attempted to incorporate historical data about the upstream and downstream area around the specific road as it can reveal hidden correlations to the traffic predictions [12].

2.4.3 3. Parameter sensitivity

As presented in [1], the high sensitivity to parameters in certain deep learning models is a topic warranting further investigation. If the parameters sensivity is high, this means that a small change in the parameters of the model can lead to a big change in the performance of the model.

The following table presents the papers that are studied for the literature review and the key highlights from each.

2.4 Key challenges identified

Author(s)	**Title**	**Key highlights**
Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014)	Short-term traffic forecasting: Where we are and where we're going	• Model selection is often based on accuracy rather than considering the characteristics of the system, such as the road. • To improve model selection, non-linear features of spatio-temporal traffic evolution and the non-stationarity of traffic should be considered. • Research predominantly focuses on hybrid neural network models, which outperform statistical models in traffic analysis.

Li, C. S., & Chen, M. C. (2013)	Identifying important variables for predicting travel time of freeway with non-recurrent congestion with neural networks	• Loop detector data from the National Freeway in Taiwan was used to model a Multilayer Perceptron. • Three factors that influence traffic flow behavior were characterized: geometric variables (slope, horizontal curve, etc.), traffic characteristics (average daily traffic, rush hours), and environmental factors (rain). • Different coding schemes for the weekday variable were tested, and encoding Monday-Sunday as 1-7 performed better than other schemes. • Adding rainfall as a feature did not improve accuracy, but including the day of the week, morning/afternoon, and historic travel time improved performance.

Mallick, Tanwi, et al. (2020)	Graph-Partitioning-Based Diffusion Convolutional Recurrent Neural Network for Large-Scale Traffic Forecasting	• Diffusion convolutional recurrent networks are effective for traffic modeling but have high computational complexity. • Graph partitioning is used to decompose a large California highway into smaller parts for individual training • Simultaneous forecasting of speed and flow is achieved using this model. • Overlapping nodes are implemented to maintain relationships between road parts.
Per Øyvind Kanestrøm (2017)	Traffic flow forecasting with deep learning	• Deep learning approach on data from NPRA • Focus on spatiotemporal data, with spatial information found to have more influence. • Models using multiple features from different stations of interest. • Experiment results showed differing performance of RNN, DNN, and SSAE.

L. Cai et al. (2020)	Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting	• Traffic Transformer is a deep learning architecture that captures the continuity, periodicity, and spatial dependencies of time series data. • Traffic Transformer outperforms baseline models such as ARIMA, historical average, LSVR, FNN, and LSTM. • Future work is needed to address the contribution of different roads at different times and further improve the model's performance.
J. Salotti et al. (2018)	Comparison of traffic forecasting methods in Urban and SubUran Context	• Evaluation of ten methods for short-term urban road traffic forecasting. • Multivariate approaches are crucial for accurate forecasting. • Nonparametric K-NN method performs best in the city center context. • Variable selection mechanisms and algorithm choice depend on road type and forecasting horizon.

Table 2.1: List of papers for literature review

Chapter 3

Theory

This section presents the general traffic flow theory and how the different variables in traffic can be related to each other. This section also explain how how different traffic-related terms are used in this paper. Then it presents a general description of traffic characteristics and the challenges it poses for modelling. Finally it presents a brief introduction to tunnel traffic and in what ways it can be different from otherwise regular traffic.

3.1 Traffic flow theory

Understanding the principles of traffic flow theory is essential in order to interpret the traffic prediction values correctly. The predicted values for traffic flow at a certain hour can be different between tunnels. However, due to their geographical specifications and road capacity the prediction must be evaluated in the right context. The dynamic and complex behaviour of traffic can be quantified by different variables, such as speed, flow and density. First, the speed of the vehicle is defined as the distance traveled per unit of time [8]. The average speed, also called the space mean speed, can be the average of aggregations by hour, day, months or so on.

3.1.1 Traffic flow

Traffic flow is the number of vehicles passed in a given frame of time, and is defined as

$$q = \frac{n}{\Delta T}$$

The movement of vehicular traffic can be classified into two distinct categories, namely uninterrupted flow and interrupted flow. Uninterrupted traffic flow is characterized by the uninterrupted movement of vehicles, which is determined only by the natural interactions between vehicles, without any external factors affecting the flow [8]. Nevertheless, certain variables such as weather conditions and time-related factors may exert an influence on this particular pattern of flow. In contrast, the regulation of interrupted flow is dependent upon a variety of factors, which include traffic signals, road infrastructure, pedestrian crossings, and other flow control mechanisms. As a result, the flow is subject to greater regulation and external influences [8].

In traffic theory, the speed, flow and density of traffic are related to each other. In an uninterrupted traffic flow scenario, the equation given below depicts this relationship [8]. This indicates that when either the density or speed is zero, the flow becomes zero. Additionally, it also portrays that a given combination of density and speed can reveal the maximum of flow for the given road. This can be seen during free flow when the traffic runs smoothly, and the density is low, and speed is high. On the contrary, during traffic congestion when the traffic flow is very low, it is caused by a high density and low speed. This relationship indicates that both the density and speed are relevant to calculate the traffic congestion in an uninterrupted traffic pattern.

Free flow Free flow is the state when traffic is flowing freely, and is below the critical density. It can be defined as

$$flow = speed * density$$

3.1.2 Traffic density

Traffic density is the count of total vehicles present per unit length of a road. Any road will have a critical density above which traffic congestion takes place. A higher value of density indicates that the vehicles are closer, while a low value showcases that the vehicles are further apart.

$$q(flow) = k(density) * v(speed)$$

Jam or critical density is when the traffic comes to complete stop. This is when the density becomes very high.

3.2 Uninterrupted flow

According to Greenshield's Model, when the traffic flow is uninterrupted the speed and density are linearly related [8]. The equation below shows the relation between speed (v), A and B (constants), and k (flow density). Values of A and B can be determined through field observations using techniques such as linear regression [8].

$$v = A - B * k$$

When combing Greenshield's assumption together with the relationship between flow, density, and speed, we can model an equation that showcases the non-linear relationship between flow and density. This can be seen in figure 3.2. A continual increase in density after the maximum flow is reached, the flow will start to decrease until jam density. This is when the flow becomes zero, and the vehicles are completely at stop [8].

Although the Greenshield model is helpful to explain the uninterrupted flow, the interrupted flow is a bit more complex and requires a deeper understanding of the dyanmics involved [8].

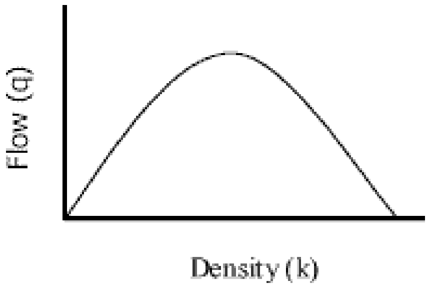


Figure 3.1: The relationship between flow and density

3.3 General traffic flow characteristics

Traffic data exhibits several characteristics that pose challenges for prediction models due to its dynamic and non-stationary nature. Firstly, traffic data is temporal, as it is time-dependent and subject to changes over time, making it challenging to predict with traditional statistical models due to the presence of dynamic underlying patterns. Moreover, traffic data also have a tendency to be spatial, as it is influenced by specific locations such as intersections or roads, and can exhibit spatio-temporal dependencies, where traffic conditions in one location at a certain time can impact traffic in nearby locations at different times.

Another challenge in predicting traffic data is its non-stationarity, as it can be influenced by external events such as festivals, weather conditions, road works, and other factors that can cause fluctuations and changes in traffic patterns. Accounting for these dynamic and non-stationary characteristics of traffic data it is crucial in developing accurate and robust prediction models in the field of traffic prediction research.

3.4 Tunnel traffic

There exist additional factors that can impact the tunnel traffic. The capacity of a tunnel is a crucial determinant, which is influenced by various factors such as the number of lanes, tunnel dimensions, and the presence of

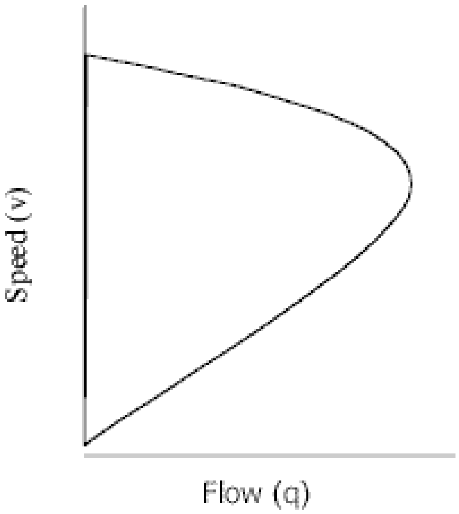


Figure 3.2: The relationship between flow and speed

roundabouts within the tunnel. Furthermore, tunnels commonly have limited entry and exit points and are subject to special regulation to control traffic congestion and uphold safety measures. The controlled environment of a tunnel may have lane control signals and speed limits that deviate from those observed on regular roads. Moreover, owing the confined structure of tunnels, incidents like accidents, breakdowns, or fire can have much more serious consequences, leading to a higher risk associated with tunnel traffic. The aforementioned risks can cause drivers to be more cautious, and even slower, compared to regular traffic.

The emission levels from from heavy vehicles have the potential to degrade the quality of air, necessitating lower speed or restricted entry to ensure safe conditions. It is worth emphazising that the lighting conditions in a tunnel can also impact the traffic flow. The shift from daylight to artificial light can have an impact on the perceptibility and response times of drivers, making tunnel traffic behavior different from that on open roads.

3.5 Terminology

This section describes the terminology used in this paper and how they are defined in this work.

- Urban: Densely populated area, where there live more than 200 persons per square kilometer (1 km x 1 km).
- Sub-urban: Less densely populated area where there live less than and equal to 200 persons per square kilometer (1 km x 1 km).
- Rush hour: Time(s) of the day during which the traffic flow reaches a maximum. Typically this is between 06.00 to 09.00 in the morning, and 15.00 to 16.00 in the evening.
- Horizon: This is the time steps in future that the model predicts for. This work finds a prediction for one step in the future, hence it is performing a short-term forecasting. The horizon depends on the granularity of the data, and the time step already present in the data.
- Tunnel tube: Directly translated from 'tunelløp' in NVDB. In this work this means a tunnel with 1 or more tubes, either in opposite direction or in the same direction.

Chapter 4

Reasearch method

This section describes the methodology that was used in the study. It uses a combined methodology that incorporates both statistical analysis and computational techniques as foundation to identify patterns, and trends in the data.

The method also applies predictive analytics, utilizing statistical algorithms and deep learning techniques to predict future values based on historical information. This includes statistical analysis, geographical understanding, and neural network capabilities to predict future values within time series data.

In this paper the area of focus for the predictions will be tunnels in Rogaland, a province in Norway on the west coast side.

A predictive model requires a comprehensive data set that showcases the traffic at different times. However, at the time of writing this thesis, there was no public data set that was prepared for only tunnel traffic in Norway. Therefore a data set is created by retrieving traffic data from two different data sources of NPRA, and joining them together through a spatial analysis.

4.1 Data foundation

This section presents the data sources used in this work.

4.1.1 Norwegian National Road database (NVDB)

Norwegian National Road database (NVDB) is open source data made available by the Norwegian Public Road Administration (NPRA) [13], under the Norwegian licence for official data (NLOD). [14]. The data is collected and maintained by the NPRA. The input to the NVDB is managed by contractors or other building companies who are doing alterations in the road network and need to report to the NPRA. NPRA ensure that the data is maintained and updated frequently and is updated according to current regulations and requirements [15]. However, NPRA does not take any responsibility of its accuracy at all times [15].

The Norwegian National road database (NVDB) is a database with information about the national road reference system. It includes among others, the main road network, consisting of the national highways, but also majority of the county and municipality roads. It also includes entities that are related to a road, for instance an accident, a tunnel, or traffic registration stations. NPRA provides an API that can be used to fetch the data, using different filters and specifications to obtain the data that is relevant for the given purpose. Full details about the NVDB can be found at [16].

4.1.2 Trafikkdata

Trafikkdata [17] is a separate database about traffic data on the Norwegian road network, and is also owned by the NPRA. The traffic registration stations consist of physical infrastructure by the road, while the traffic registration points (found in Trafikkdata) are the actual physical location on the road where the data is captured [17].

Traffic data is collected from inductive loop sensors installed on the road. When a vehicle drives over them, it captures information such as its length, speed, vehicle class, distance to the preceding vehicle, lane of travel, and direction [18]. Cars overtaking or driving between lanes are likewise handled by the system [18].

Traffic data is used to fetch the information about the different sensors that are located on the roads, and the corresponding hourly traffic flow per traffic registration point.

The traffic flow data is available for the public, however the related speed data is restricted and requires special access [19]. Therefore, only the hourly aggregated traffic flow will be used in this thesis.

4.1.3 Unknown link between NVDB and Trafikkdata sensors

A set of objects in NVDB are the traffic registration stations which serve as the primary monitoring stations for nearby registration points (sensors). These stations are not the physical sensors where the data is collected, and must be considered separately from the points. Secondly, the NVDB also includes another set of objects known as traffic detectors (NVDB-id:167). However, upon performing a geo analysis of the stations and detectors together with the sensors from Trafikkdata, it comes clear that they do not share the exact same geographical location. Therefore it is worth highlighting that as of the time of writing this thesis, there is no established relationship between the traffic registration stations in NVDB, and the registration points in traffic data. This relation is necessary in order to find which sensor is close to or inside a tunnel. However, both data sources serves the geographical coordinates that will be used in this work to find the relation between tunnels and points.

4.1.4 Statistics Norway (SSB)

Statistics Norway is the national institute of Norway responsible for producing official statistics [20]. In addition to official statistics, SSB provides several statistics by geographical location, known as grid maps [21]. Population count for the different areas of the country can be found through such a grid map called the 'Population Statistics on grid' and can be found at [21]. This map is added to QGIS as a separate layer by connecting to [21] through the WFS service. The population data is used to evaluate whether a tunnel is urban or not.

4.2 Data retrieval

This section describes how the data was retrived from the different sources.

1. Fetch tunnel tubes from NVDB through API
2. Fetch traffic registration points from NVDB through API
3. Find points that are near a tunnel using spatial join
4. Find categories of tunnel tubes using GIS-tool

4.2.1 NVDB Rest API

The Norwegian Public Roads Administration (NPRA) provides an interface to the Norwegian Road Database(NVDB) via a Representational State Transfer (REST) application programming interface (API) [22]. The data from the NVDB is retrieved using the NVDBAPI-v3 Python library. This library is maintained by Chief Engineer Jan Kristian Jensen at NPRA and can be found at [23]. The library includes two main objects: 'NVDBFagData' which are objects that are related to a road network such as a tunnel. The other is the 'NVDBVegNett' which is the road network itself and is formed with different road segments attached to each other through links. The documentation for the API can be found at [24] and the GitHub repository with the python library is given at [23].

4.2.2 Trafikkdata API

The Trafikkdata database is a GraphQL database, and the web interface to their API can be found at [25]. The connection to this API is by using the requests library for python.

1. Get traffic registration points

The query sent in through the POST request is a GraphQL query inspired by the query example showcased at [25], but amended for the need of this work. The query can be seen at listing 4.1 and can be found in the get_data.py file. The following parameters are added to the query:

- IsOperational: true
- TrafficType: VEHICLE
- RegistrationFrequency: Continuous
- CountyNumbers: 11
- fromAccordingToRoadLink
- Location coordinates in latitude and longitude

The query is fetching the id, name, direction and geographical location for the different traffic registration system that is present in Rogaland (province id: 11). This province id is the same as in NVDB. The different ids for the provinces can be found at https:/nvdbapiles-v3.atlas.vegvesen.no/omraderfylker.

2. Get hourly traffic for a given traffic registration point

A different graphQL is constructed to fetch the hourly traffic data flow through the API, or the data can be downloaded directly from their web interface. The data exported from the web interface is further preprocessed and prepared for analysis. However, during development the API was not able to fetch large amounts of data. Therefore by the end of the thesis, the web interface was used for hourly data export.

```
{
  trafficRegistrationPoints(searchQuery:
  {isOperational: true, trafficType: VEHICLE, ...
      registrationFrequency: CONTINUOUS, ...
      countyNumbers:11}) {
    id
    name
    direction {fromAccordingToRoadLink}
    location {
      coordinates {
        latLon {
          lat
          lon
        }
      }
    }
  }
}
```

Figure 4.1: GraphQL query to fetch traffic registration points from Trafikkdata.no APi

4.3 Data integration with geo-spatial analysis

This section presents the steps taken to add the different data sets into the GIS-tool, and how the geo-spatial analysis is performed..

1. Adding Layers in GIS Tool with Python and QGIS
2. Find registration points near the tunnels using spatial join & nearest neighbours
3. Define urbanization of tunnel through spatial intersection

4.3.1 Adding layers in GIS Tool with Python and QGIS

Each data source is added as a separate layer in the QGIS. Through the Python library NVDBAPI-V3 mentioned earlier, there comes separate files

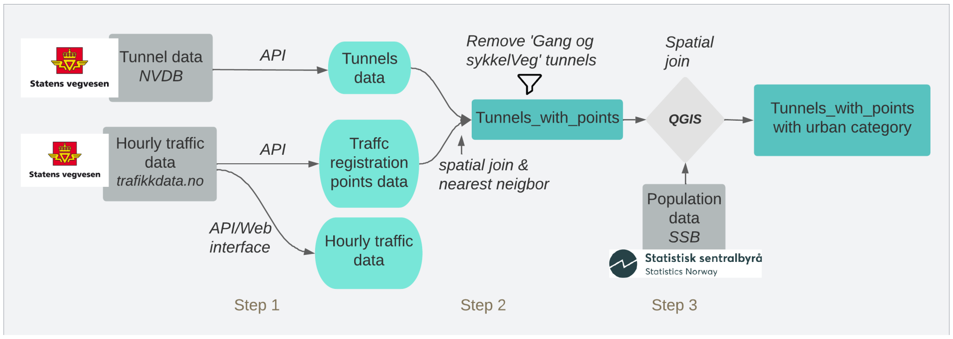


Figure 4.2: Diagram showing the steps taken to fetch the data

that allow integration of NVDB with QGIS. This library has copyright (c) by Jan Kristian Jensen and is available under the MIT lisence. The tunnel tubes are fetched from the NVDB using road object id = 67 with a filter set on province = 11 for Rogaland. The population grid from SSB is added through the WFS service, and the open street MAP is added through inbuilt functionality in QGIS.

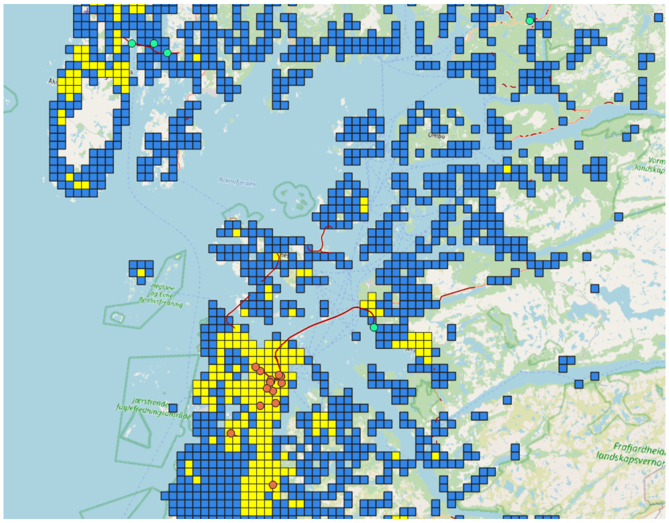


Figure 4.3: Map in QGIS with tunnels (red), urban points (orange) and suburban points (turqoise)

4.3.2 Find registration points near the tunnels using nearest neighbours

A graphQL query is sent to Trafikkdata to fetch the traffic registration points and their id, name, and most importantly its coordinates. Several points are not located close to a tunnel, or there may be an cross-sectioning road between the tunnel entrance and the traffic registration point. The target was to find the points that are located close to a tunnel without any crossing road in between, which can give us the traffic that is either going in/out of a tunnel or through a tunnel. This is achieved by performing a spatial join between the points spatial coordinates, and the tunnels. The spatial join is performed using the sjoin_nearest() method of Geopandas library. This function performs a spatial join of the both to find overlap, and is given a distance within which it can look for neighbouring tunnels and points. The distance is set på 0.002. This find the points that are very near the tunnels geometry. The drawback here is that if the tunnel is underwater, or a bridge, the registration points that are above or under those in real, will appear as if thery are in the same place. However, this is a limitation as this work is being performed in a 2D space. Finally, the resulting geo dataframe is written to to file.

The points returned by the API are converted from latitude and longitude into a geometric coordinate, and converted to the right coordinate reference system (crs).

```
gdf = gpd.sjoin_nearest(tunnels, points, max_distance=dist)
```

4.3.3 Define category of tunnel through spatial intersection

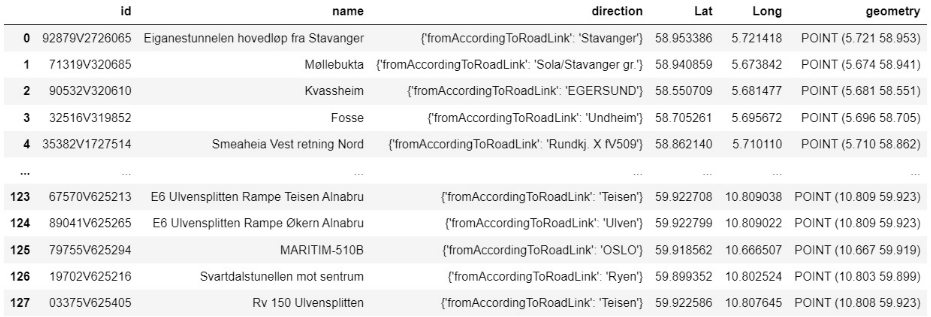
The following steps are performed to categorize the tunnels as urban or sub-urban.

- Select from the population grid from SSB only the grids that have a population above 200 per square km
- Add the data to a geo dataframe and create a geometry column

- Read in data from file that has tunnels and their nearest points
- Loop through each points and check if it is intersecting with the selected population grid
- Write the resulting dataframe to file

This is achieved using the functions defined in the get_category_of_tunnels.py file through the python console in QGIS. First, the relevant population grids are selected. In this work a value of 200 is set, so only grids where the population is above 200 per square km are selected. The file with the nearest points and tunnels is read into a dataframe, and looped through to see if the geometry of the points is intersecting with the grids. If a point is intersecting within the grid or the polygon geometry, the point is categorized as urban, otherwise sub-urban.

The subset of the resulting dataframe is presented in figure 4.4.



	id	name	direction	Lat	Long	geometry
0	92879V2726065	Eiganestunnelen hovedløp fra Stavanger	{'fromAccordingToRoadLink': 'Stavanger'}	58.953386	5.721418	POINT (5.721 58.953)
1	71319V320685	Møllebukta	{'fromAccordingToRoadLink': 'Sola/Stavanger gr.'}	58.940859	5.673842	POINT (5.674 58.941)
2	90532V320610	Kvassheim	{'fromAccordingToRoadLink': 'EGERSUND'}	58.550709	5.681477	POINT (5.681 58.551)
3	32516V319852	Fosse	{'fromAccordingToRoadLink': 'Undheim'}	58.705261	5.695672	POINT (5.696 58.705)
4	35382V1727514	Smeaheia Vest retning Nord	{'fromAccordingToRoadLink': 'Rundkj. X fV509'}	58.862140	5.710110	POINT (5.710 58.862)
...	...	...	...	...	...	...
123	67570V625213	E6 Ulvensplitten Rampe Teisen Alnabru	{'fromAccordingToRoadLink': 'Teisen'}	59.922708	10.809038	POINT (10.809 59.923)
124	89041V625265	E6 Ulvensplitten Rampe Økern Alnabru	{'fromAccordingToRoadLink': 'Ulven'}	59.922799	10.809022	POINT (10.809 59.923)
125	79755V625294	MARITIM-510B	{'fromAccordingToRoadLink': 'OSLO'}	59.918562	10.666507	POINT (10.667 59.919)
126	19702V625216	Svartdalstunellen mot sentrum	{'fromAccordingToRoadLink': 'Ryen'}	59.899352	10.802524	POINT (10.803 59.899)
127	03375V625405	Rv 150 Ulvensplitten	{'fromAccordingToRoadLink': 'Teisen'}	59.922586	10.807645	POINT (10.808 59.923)

Figure 4.4: Geo dataframe with details about the registration points

In addition, the points are converted to POINT objects using Shapely library. Population grids are converted to POLYGONS.

The returned object returns also Gang og sykkelveg, and sykkeveg which are filtered out for the sake of this work.

Chapter 5

Data analysis

Time series analysis establishes a robust and scientific basis for utilizing temporal data, augmenting both the theoretical understanding and practical applications of deep learning methodologies. This section presents the data analysis performed on the acquired data set.
The following are the questions that are the basis for this analysis:

- What is the trend and seasonality pattern of tunnel traffic?
- Does the technical features of a tunnel relate to its daily traffic?
- Investigate the impact of urban/sub-urban feature of a tunnel on traffic flow

The data utilized in this analysis is exclusively derived from Lane 1 for each tunnel, for the year 2021 and the scope of our analysis encompasses the following tunnels:

- Storhaugtunnelen, Urban, 1275 m
- Auglendtunnelen, Urban, 390 m
- Iglatjørntunnelen, Suburban, 447 m
- Kleppetunnelen, suburban, 515 m

5.1 Data preprocessing

1. First select that only one side of the road is shown, this is so that the data is not aggregated because both the points have the same name 2. The roads are named as 1, 3, and 2 and 4. So i select only 1 and 3. 1 and 2 is for regular, but for tunnels that have more than 1 lane that lane is named 3 and 4 respectively in each direction. 3. The data from the hourly trafikkdata is used for feature engineering of the data 4. The data is added a DateTimeIndex so its easier to perform resampling.

This data includes the hourly traffic for both lanes in a tunnel, respectively for lane 1 and 2 going in opposite direction. For tunnels that are tube tunnels, and have more than one lane, are numbered 3 and 4 respectively. The data from this file is separate into a dataframe with data for lane 1 and 3, and for lane 2 and 4. This separation ensures that the traffic patterns of both directions are not overlapped and disturb the actual traffic patterns.

The tunnels that are needed are downloaded from the trafikkdata web interface. The resulting data consists of several fields, where the following fields are selected for this thesis:

- Traffic Registration Station ID
- Traffic Registration Station Name
- From
- To
- Lane
- Traffic flow
- Degree of coverage

Handling missing values

The data did not contain a lot of missing values. However, some data points were labeled with a - for missing values for traffic flow for that hour.

This could be due to issues with the sensor, during data transmission to the database, or because the count was so low that the data had to be eliminated due to GDPR. In the selected tunnels, there were very few missing values. Ideally, an informed choice must be made to select the best technique to fill in missing values.

Data scaling

The majority of the tunnels follow a similar pattern in terms of seasonality and trend. However, the amount of traffic in the tunnels is the factor that differentiates them most. Some tunnels like Auglendtunnelen or Austrått have very high levels of traffic, compared to for instance Iglatjørntunnelen. The RobustScaler from sklearn is used to scale the data, as this is not sensitive to extreme values. This scales the data based on statistics such as the median and interquartile range.

5.2 Data exploration

The graph in figure 5.1 is showing the total traffic flow per 6 hours, for one week. Each tunnel in the sample follow a more or less similar pattern, with two local maximums each day on the weekdays and one maximum on Saturday and Sunday. However, the magnitude of the traffic flow for each tunnel, is very different. Additionally, the tunnel with the highest volume also has the highest change between minimum and maximum during the day. It can also be noted that the Iglatjørntunnelen has very less traffic compared to the others.

When inspecting more in depth into the daily trend of the tunnels, two daily maximums are seen. This can be seen in figure 5.2. The first peak depicts the morning rush hour, while the second one showcases the mid day rush hour. However, the Iglatjørntunnelen does clearly follow this trend. This can be due to this tunnel not being surrounded by many with the typical 9-5 working hours.

The graph in figure 5.3 shows the trend based on the day of the week. Based

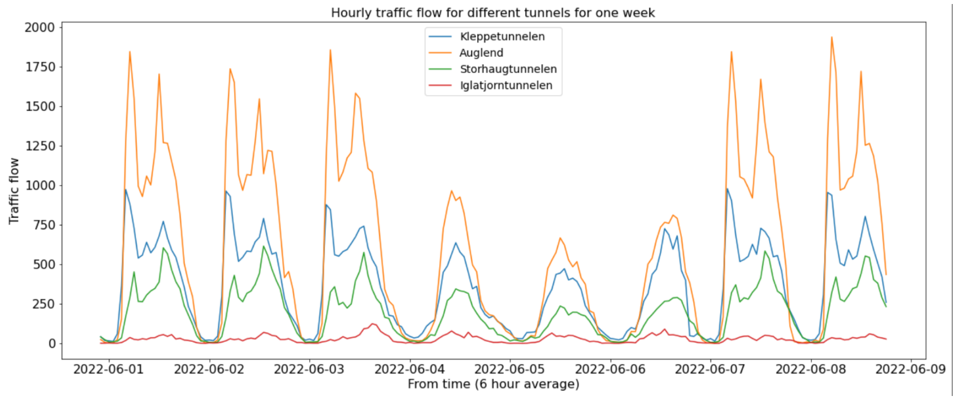


Figure 5.1: Total traffic flow per 6 hours for different tunnels in one week

on the given sample, the data indicates a difference between the traffic pattern between the weekday Monday to Friday, and weekend. During the weekend the increase in traffic is on average later than the rest of the week, and has a smooth increase and decrease during the day.

Figure 5.5 indicates that for sub urban tunnels, the traffic volume shows an increase in the weekend in contrast to urban tunnels. The urban traffic tends to show a stronger seasonality compared to the urban traffic. This could potentially be related to the holidays, and people tend to travel to norwegian suburbs for holidays. Additionally, the figure in 5.4 shows the weekly sum of traffic volume based on the tunnel category (urban/sub-urban). The visualization suggest a high variability during summers.

5.3 Feature engineering

As stated in the literature review - the spatiotemporal features seems to be the most important features that has helped models perform better.

In the context of time series data, feature extraction is extracting useful information from timestamps in order to improve the data's understanding and predictive power.One method for extracting features from time series data is to use the DateTimeIndex capability in the pandas library and the datetime module in Python. This enables access to different characteristics

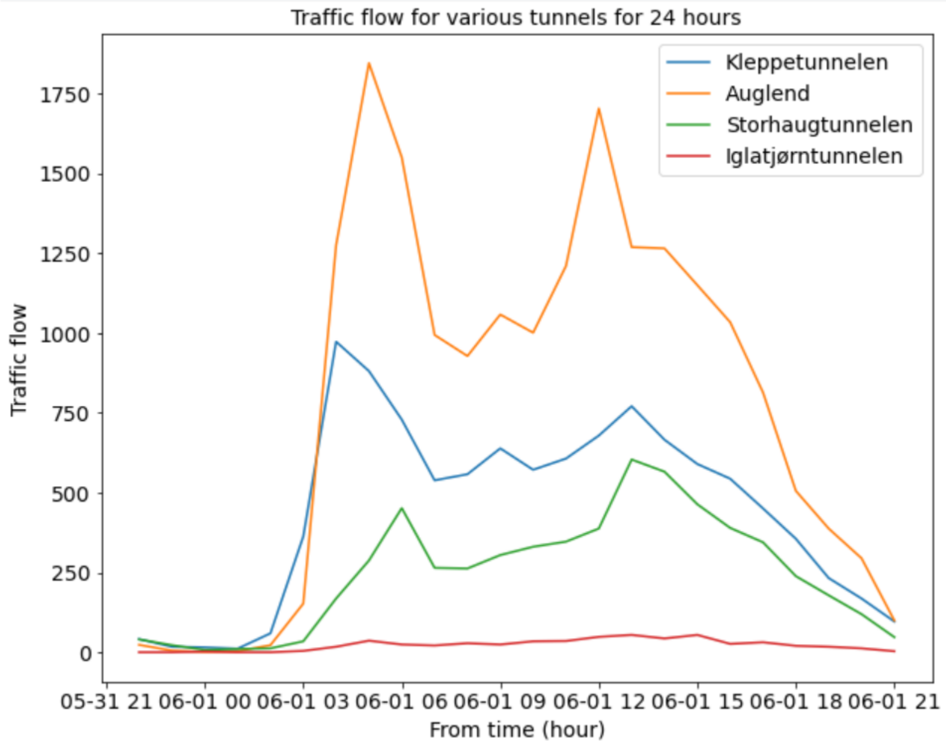


Figure 5.2: Hourly traffic in a day for various tunnels in Rogaland

and methods to extract relevant features by transforming the timestamp data into a DateTimeIndex. The DateTimeIndex extracts features such as the day of the month, month and day name (e.g., Monday, Tuesday), and hour.

The NVDB tunnel data have their names, lengths, width, and several other metadata about the tunnels. The resulting data set included different types of tunnels. However, the following types were removed. tunnels = tunnels[(tunnels["typeVeg"] != 'Gang- og sykkelveg') & (tunnels["typeVeg"] != 'Sykkelveg')]

Using the hourly traffic data, a summation of the yearly traffic for each tunnel is computed. For example, the total yearly traffic volume for Austrått tunnel is 3 021 000 cars/year. This gives an idea of the difference between the volumes for different tunnels.

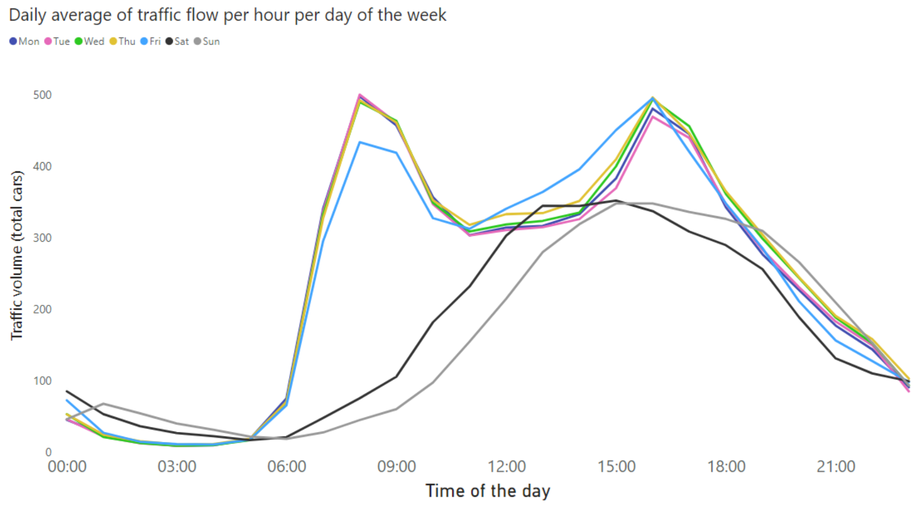


Figure 5.3: Hourly traffic flow in a day for every day of the week

5.4 Correlation matrix

The correlation heat matrix is produced based on the data from the tunnels fetched from NVDB. The features that are collected are displayed in a matrix in order to see whether any features are related to each other, and in particular to the total traffic quantity. The correlation matrix shows that there are some linear relationships present between the length of the tunnel and the opening year, the tunnel profile and width, type of road, and width. Additionally, a linearity between the type of tunnel and category is present. When inspecting data further there is a special type of tunnel that is present mostly in urban areas which is the cut and cover type (lokk tunnel). The yearly traffic (vol_by_thousand) is slightly correlated to municipality (kommune), category and road number (vegnummer).

5.5 Summary statistics

Table 5.8 shows the summary statistics of all tunnels that are fetched from NVDB. This particular visualizatin is performed on all tunnels retriavable

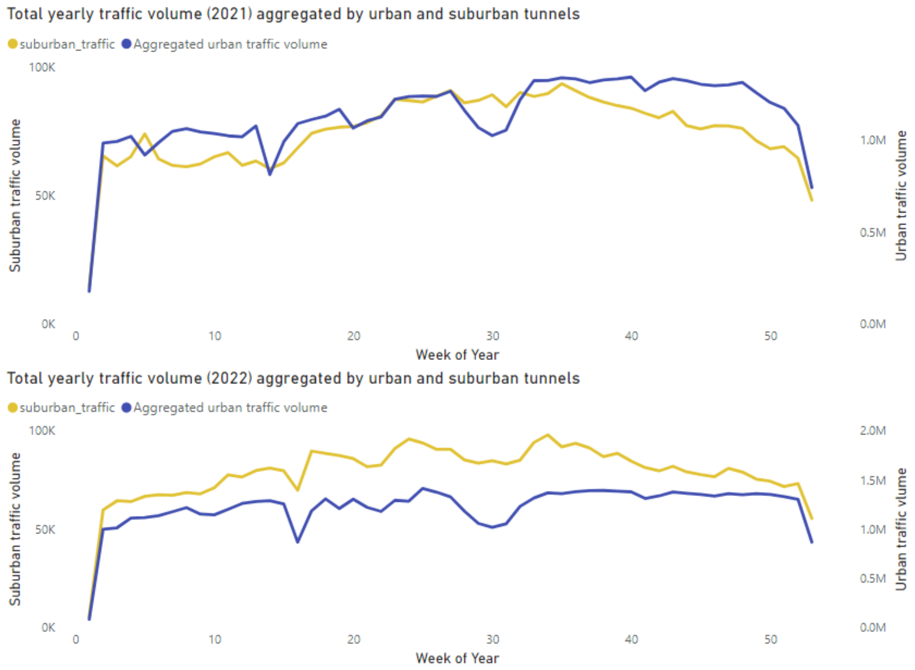


Figure 5.4: Aggregated weekly sum of traffic volume for urban and suburban

from NVDB. The count shows the number of rows present in this dataframe, however in this data set there are 1447 unique tunnels. The duplicates occur if there are both canalized road and normal road present in the tunnel, or if the tunnel is separated into several parts. The width can be seen is present only for 1123 tunnels, while the height is only for 376. The strong correlation of width in the correlation matrix can be related to the missing values.

In Trafikkdata only one point is present in the nearby location of a tunnel. Therefore when a spatial join is performed, the same registration point can be merged with three different tunnels in NVDB, as is shown in figure 5.9.

Finally, it is also worth mentioning that 2021 summer has a higher traffic towards suburbs in summer could be due to the corona virus pandemic. The two major seasonal factors for both traffic patterns are easter and summer holidays. However, this must be further investigated using more tunnels and more data.

5.5 Summary statistics

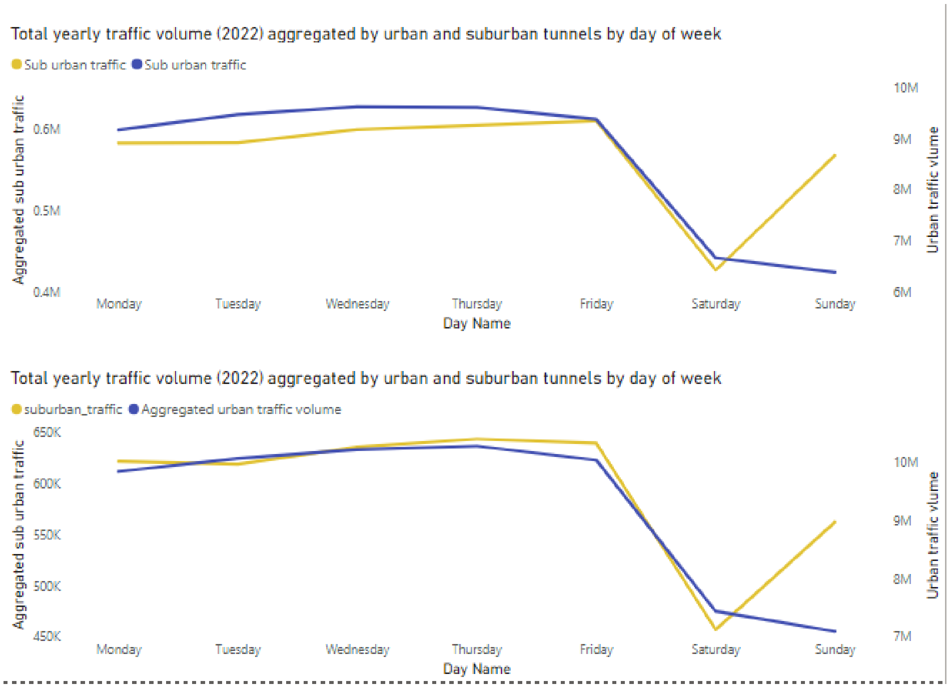
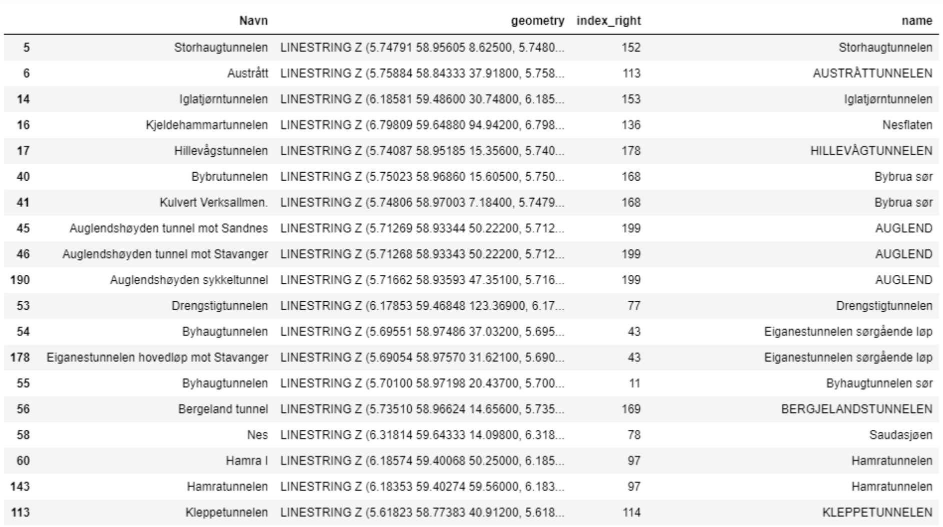


Figure 5.5: Total yearly volume for urban and sub urban tunnels aggregated per week day



	Navn	geometry	index_right	name
5	Storhaugtunnelen	LINESTRING Z (5.74791 58.95605 8.62500, 5.7480...	152	Storhaugtunnelen
6	Austrått	LINESTRING Z (5.75884 58.84333 37.91800, 5.758...	113	AUSTRÅTTUNNELEN
14	Iglatjørntunnelen	LINESTRING Z (6.18581 59.48600 30.74800, 6.185...	153	Iglatjørntunnelen
16	Kjeldehammartunnelen	LINESTRING Z (6.79809 59.64880 94.94200, 6.798...	136	Nesflaten
17	Hillevågstunnelen	LINESTRING Z (5.74087 58.95185 15.35600, 5.740...	178	HILLEVÅGTUNNELEN
40	Bybrutunnelen	LINESTRING Z (5.75023 58.96860 15.60500, 5.750...	168	Bybrua sør
41	Kulvert Verksallmen.	LINESTRING Z (5.74806 58.97003 7.18400, 5.7479...	168	Bybrua sør
45	Auglendshøyden tunnel mot Sandnes	LINESTRING Z (5.71269 58.93344 50.22200, 5.712...	199	AUGLEND
46	Auglendshøyden tunnel mot Stavanger	LINESTRING Z (5.71268 58.93343 50.22200, 5.712...	199	AUGLEND
190	Auglendshøyden sykkeltunnel	LINESTRING Z (5.71662 58.93593 47.35100, 5.716...	199	AUGLEND
53	Drengstigtunnelen	LINESTRING Z (6.17853 59.46848 123.36900, 6.17...	77	Drengstigtunnelen
54	Byhaugtunnelen	LINESTRING Z (5.69551 58.97486 37.03200, 5.695...	43	Eiganestunnelen sørgående løp
178	Eiganestunnelen hovedløp mot Stavanger	LINESTRING Z (5.69054 58.97570 31.62100, 5.690...	43	Eiganestunnelen sørgående løp
55	Byhaugtunnelen	LINESTRING Z (5.70100 58.97198 20.43700, 5.700...	11	Byhaugtunnelen sør
56	Bergeland tunnel	LINESTRING Z (5.73510 58.96624 14.65600, 5.735...	169	BERGJELANDSTUNNELEN
58	Nes	LINESTRING Z (6.31814 59.64333 14.09800, 6.318...	78	Saudasjøen
60	Hamra I	LINESTRING Z (6.18574 59.40068 50.25000, 6.185...	97	Hamratunnelen
143	Hamratunnelen	LINESTRING Z (6.18353 59.40274 59.56000, 6.183...	97	Hamratunnelen
113	Kleppetunnelen	LINESTRING Z (5.61823 58.77383 40.91200, 5.618...	114	KLEPPETUNNELEN

Figure 5.6: Table

5.5 Summary statistics

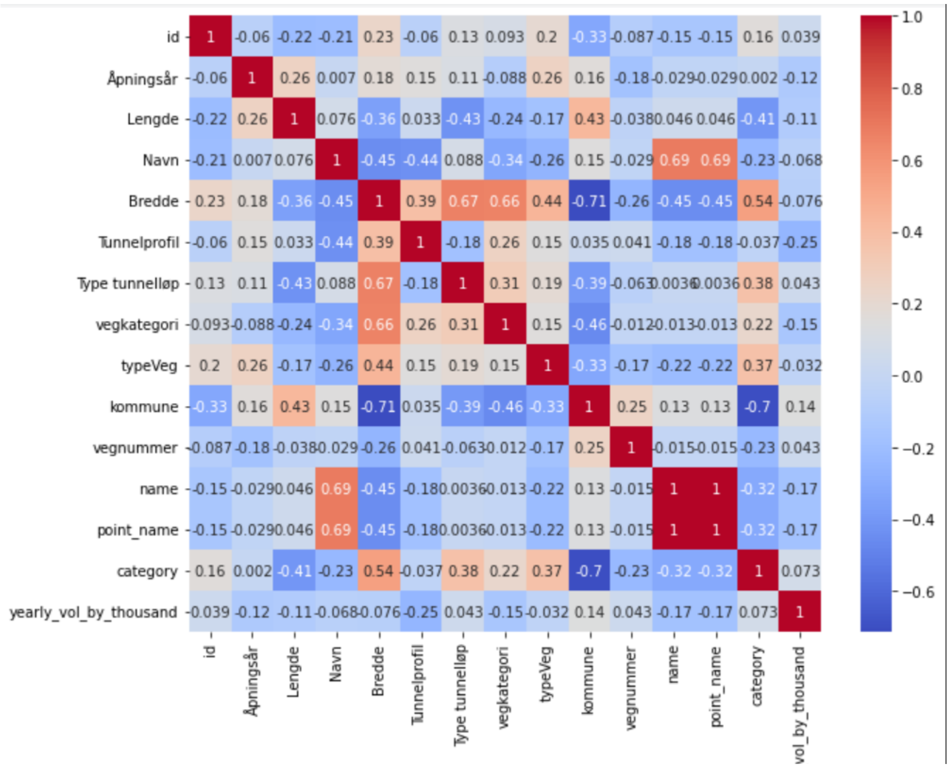
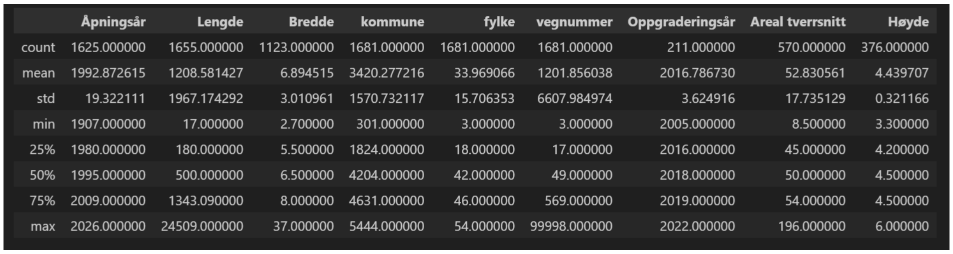
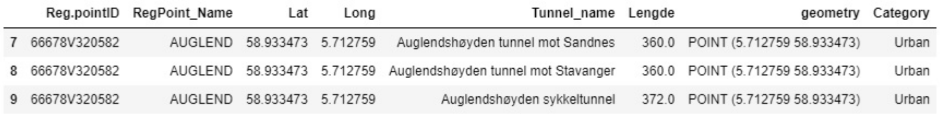


Figure 5.7: Correlation heat matrix



	Åpningsår	Lengde	Bredde	kommune	fylke	vegnummer	Oppgraderingsår	Areal tverrsnitt	Høyde
count	1625.000000	1655.000000	1123.000000	1681.000000	1681.000000	1681.000000	211.000000	570.000000	376.000000
mean	1992.872615	1208.581427	6.894515	3420.277216	33.969066	1201.856038	2016.786730	52.830561	4.439707
std	19.322111	1967.174292	3.010961	1570.732117	15.706353	6607.984974	3.624916	17.735129	0.321166
min	1907.000000	17.000000	2.700000	301.000000	3.000000	3.000000	2005.000000	8.500000	3.300000
25%	1980.000000	180.000000	5.500000	1824.000000	18.000000	17.000000	2016.000000	45.000000	4.200000
50%	1995.000000	500.000000	6.500000	4204.000000	42.000000	49.000000	2018.000000	50.000000	4.500000
75%	2009.000000	1343.090000	8.000000	4631.000000	46.000000	569.000000	2019.000000	54.000000	4.500000
max	2026.000000	24509.000000	37.000000	5444.000000	54.000000	99998.000000	2022.000000	196.000000	6.000000

Figure 5.8: Summary statistics of tunnel_with_points dataframe main features



	Reg.pointID	RegPoint_Name	Lat	Long	Tunnel_name	Lengde	geometry	Category
7	66678V320582	AUGLEND	58.933473	5.712759	Auglendshøyden tunnel mot Sandnes	360.0	POINT (5.712759 58.933473)	Urban
8	66678V320582	AUGLEND	58.933473	5.712759	Auglendshøyden tunnel mot Stavanger	360.0	POINT (5.712759 58.933473)	Urban
9	66678V320582	AUGLEND	58.933473	5.712759	Auglendshøyden sykkeltunnel	372.0	POINT (5.712759 58.933473)	Urban

Figure 5.9: Three different tunnel names merge with the same registration point

Chapter 6

Experiments

The following tunnels are used in the experiments. The data sets are downloaded directly from Trafikkdata.no web interface and prepared in Microsoft Excel by removing unnecessary columns, renaming columns and adding categorical feature. However, this can be automated and created using LabelEncoder by time opportunity.

The data is from 01 June 2022 until 01 June 2023. The experiments are based on the same model in the next section. The intention is to create different combination of tunnels, and features to see how the performance changes.

The following tunnels are used in these experiments:

- Sub-urban tunnel 1: Iglatjørntunnelen using registration point 93763V320622
- Sub-urban tunnel 2: Drengstigtunnelen using registration point 41663V319808
- Sub-urban tunnel 3: Nesflaten using registration point 51143V319682
- Urban tunnel 1: Storhaugtunnelen using registration point 57279V320244
- Urban tunnel 2: Auglendtunnelen using registration point 66678V320582
- Urban tunnel 3: Kleppetunnelen using registration point 72379V1688678

The following experiments are performed.

1. Train RNN model on a all urban tunnel, and evaluate its performance
2. Train RNN model on only Storhaugtunellen with temporal features and evaluate its performance
3. Train RNN model on all six tunnels
4. Train RNN model on all six tunnels with category feature
5. Train RNN model on all six tunnels with category feature and temporal features

In experiment #1 the model is trained on only urban tunnels. The purpose of this is to see how well the model performs on sub-urban tunnels. The model's performance on sub-urban tunnels can provide an indication of how distinct both categories are. If the performance is good, it can indicate that the underlying pattern for both categories are not so different.

In experiment #2 the model is trained only on Storhaugtunnelen, but with temporal features. The temporal features DayNumber, Hour, Month, DayOfMonth are extracted from the given time stamps in the data set.

In experiment #3 the model is trained all six tunnels. The goal of this experiement is to see how well the model is able to generalize when it has seen both types of tunnels. However, no other feature is given, and the model is only learning from the inherent temporal dependencies in the data.

In experiment #4 the model is trained on all six tunnels, together with the category features. The performance gain can reveal the significance of this feature, and also the accuracy of the category feature as it is created in this thesis. If this feature provides a good improvement, it may indicate that the spatial location of the tunnel is of high significance.

In experiment #5 the model is trained on all six tunnels, together with the category and temporal features. In this experiment, both types of features, and all tunnels are added to evaluate the performance gain.

Chapter 7

Model

This section describes the deep learning approach - the recurrent neural network model that is implemented in this thesis, and the baseline model that is used for comparison. First the baseline model is described, followed by a description of the RNN model. Finally a description of the model implementation is presented.

The RNNs model is implemented using Tensorflow, while also using several other useful libraries such as Numpy, and Pandas.

7.1 Baseline model

The baseline approach is a simple memory less model that is used to evaluate relative performance against the RNN model. This model is a naive approach where the predicted value is equal to the previous value. That implies that in a set of consequent time series values, if the last known value is y_t then the next value in the sequence is $y_{t+1} = y_t$.

The evaluation metrics RMSE and MAE are utilized for evaluation, and the equation for each can be seen in equation 7.1 and 7.2. These calculations use the difference between two consequent values.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y[i] - y[i-1])^2} \tag{7.1}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y[i] - y[i-1]| \tag{7.2}$$

7.2 Recurrent Neural Network

RNNs were originally developed for natural language processing, but has paved its way into time series in the recent years. The literature review suggested that RNNs are well adapted to traffic forecasting, particularly in terms of simulating sequential data. As mentioned in in the introduction, traffic data is a time series by definition which implies that it has temporal dependencies. The traffic flow at a given period may have dependencies on traffic flow in the past. The literature review also revealed that RNNs have been a deep learning model that has outperformed others in terms of traffic forecasting.

RNNs, are built to deal with time dependencies. They accomplish this by retaining a form of 'memory' about prior inputs in the sequence via hidden state vectors, allowing them to remember previous information. Furthermore, RNNs can describe non-linear correlations in data that traditional statistical methods may fail to capture. RNNs may learn and improve with new data over time, which corresponds to the dynamic nature of traffic patterns.

The general equation for the output for each hidden layer in a simple RNN is as shown in equation 7.3.

$$h^j(k) = \sigma_h(W_{j-1}h^{j-1}(k) + U_j h^j(k-1) + b_c) \tag{7.3}$$

- $h(W_{j-1}h^{j-1}(k)$ is the ouput from hidden layer j

- k is the time step
- W_{j-1} and U_j are the input and recurrent matrices

The general equation for the output of the prediction layer of a simple RNN is as shown in equation 7.4

$$\hat{y}(k) = W_L h^L(k) \tag{7.4}$$

- $\hat{y}(k)$ is the predicted value the output layer produces
- W_L is the output matrix

Generally, for RNNs different loss functions may be utilized. In this work the mean squared error is used during model training to adjust the weights between the neurons. The mean squared error is a suitable options for a simple RNN performing a regression task. It is computationally efficient to work it, it also has the ability to penalize larger error. That implies that if the predicted traffic flow is far off, it will be quantified significantly in this measure.

$$MSE = \sum_{k=0}^{m} \|y(k) - \hat{y}(k)\|^2 \tag{7.5}$$

Similar to the baseline model, the same evaluation metrics are utilized for the RNN. In this case, the difference between the actual and predicted value is used to calculate how far oss the prediction is. The formula for root mean squared error can be seen in equation 7.2 and for the mean absolute error can be seen in equation 7.2

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \bar{y}|$$

7.3 Model implementation

This section presents the steps and preparations performed on the data set and the implementation of the simple recurrent neural network that is utilized in this thesis.

7.3.1 Data scaling

As presented in section 5 the traffic flow for different tunnels possess a different magnitude of fluctuations, meaning some tunnels are highly busy compared to other. Therefore the data needed to be scaled to normalize the data set into a given range. During development several scalers from the sklearn library were attempted. However, the RobustScaler was the one that was able to squeeze the amplitudes of the tunnels into the given range. Importantly, because the model is designed to take as input both single or multiple tunnels,it is important that the same scaler is used across all tunnel features. Therefore the features for all tunnels are concatenated together, before the RobustScaler is applied. Performing the scaling separately for each tunnel would not normalize the values within similar range due to the tunnels having different maximum flows. Similarly, the features and target are scaled with a separate instance of the scaler. The scaled data is then split into train and test.

7.3.2 Validation set

The validation set is used actively during development to monitor the model performance. This set is given to the model during training. However, it is not able to train on this data, but rather evaluate its performance on unseen data during training. The purpose of the validation test is to identify how well the model is able to generalize to unseen data, and is used as an indicator to monitor when to stop the training to prevent overfitting. The validation and testing set are selected to be the last portions of the the data set to respect the integrity of time series. The given data set is from 01. June 2022 to the same day the year after, the testing is performed on the last four weeks of the data set.

7.3.3 Creating sequence samples

The data set is divided into sequences of consequent values. The sequence length is set to two weeks in this work. However, more elaborate testing can be performed to determine the optimal sequence length. This can given an indication of how far in history the data is time-dependent. These sequences of 2 weeks and the selected number of features creating a matrix of features x sequence_length. Furthermore, these are then divided into batches that are served to the model. The usability of creating batches is that they can be processed in parallel and can potentially speed up the training process. The batch size is how much of the data is presented to the model at one time and is able to adjust the weights using the mean squared error between the actual and predicted values. A smaller batch size mean that the model updates the weights more frequently and the learning process may be faster. However, such frequent updates may introduce noise into the model.

7.3.4 Create model

The RNN is created such that different hyperparameters can be chosen by sending them in as parameters to the create_model() functino, as seen in listing given below. This is an example of how the model can be trained by inserting different parameters and hyper parameters. The intention remains that the by changing these parameters and monitoring the change in performance, an optimal combination of these can be achieved.

```
model = create_model(MODEL_TYPE=model_type, ...
    layers=number_layers_NN, ...
    units_num=number_units_per_layer_NN, ACT="tanh", ...
                    ...
    loss_func="mean_squared_error",X_train=X_train_datasets, ...
    y_train=y_train_datasets, epochs=epochs, ...
```

```
      batch_size=X_train_datasets.shape[0], ...
      X_val=X_val_datasets,y_val=y_val_datasets, ...
      shuffle=False, ret_seq=True,dropout_rate=0.1, ...
      learning_rate=0.001, optimizer="adam", ...
                        ...
      early_stopping_pmt=1,_Flag_time_features=Flag_time_features,
  _Flag_category_features=Flag_category_features)
```

7.3.5 Features

In this thesis two types of features are used in the model to evaluate the performance. That includes the temporal features Day Number, Hour, Month, and Day of Month, extracted from the 'To' timestamp. These features are added to see if the seasonal or cyclical patterns can be detected by the model using these features. The other feature that is used is the category that is obtained during the geo-spatial analysis presented in section 4.3.

The combinations of adding different features and amount of tunnels for training, is controlled by using a set of boolean variables.

- Flag_all_datasets: Set to 'True' if predictions are performed on all tunnels
- Flag_category_features: Set to 'True' if the the category feature is included
- Flag_time_features: Set to 'True' if the temporal features DayNumber, Hour, Month are used. DayOfMonth extracted from the "To" timestamps are included

The create_model() function uses these boolean variables to handle the input from a single tunnel or from multiple tunnels. In the beginning, the code uses the Flag_time_features and Flag_category_features variables to figure out which features to use. The first one means that only the temporal features are used by the model, while the second one means that the category feature (urban/sub-urban) is used. This can be advantageous when experimenting with different combinations of features to see which one provides the largest performance gain.

7.3.6 Input layer

The input to the model is created such that the model can be trained on a single tunnel or multiple tunnels having the same amount of data points. This means that the model is able to handle a dynamic shape of the input data, where the dimensions can be changed by the sequence_length and the number of features used.

In terms of neural networks, deep architectures can learn more advanced patterns compared to shallow neural networks [26]. However, with the hypothesis that tunnel traffic may not be as complex as otherwise regular traffic is, a shallower network is trained rather than a deeper in terms of layers.

The following hyper-parameters are used to train the simple RNN model:

Hyper parameter	**Value**
Layers	2
sequence_length	336 hours
Activation function	tanh
neurons per layer	128
loss function	mse
optimizer	adam
learning rate	0.001
time_step	1 hour
epochs	400

The number of layers, epochs and neurons are set to 2, 400 and 128 where there is 1 input layer, 1 hidden layer and 1 output layer. The values are based on the model's performance on the validation set. Using the validation set to monitor how the validation loss is behaving together with the training loss, gives an indication of which hyper parameters can be a good combination. However, the search for the optimal combination is not extensive.

The time horizon used in this model is set to 1, due to the time step already present in the data.

7.3.7 Addressing overfitting

Several techniques are added in order to prevent overfitting of the model to the training set. Each layer is followed by a drop out layer which implies that it stochastically eliminates a portion of the inputs during the training phase. This is monitored by the drop_out rate, which is the fraction of features that are nullified. This techniques regulates the training of the model and forces it to learn more robust features that may be useful.

The other technique that is added is early stopping. This allows the model to stop the training earlier if the performance on the validation set does not improve. Early stopping can save computational resources. If early_stopping_pmt is set to 1, it uses an "early stopping" mechanism that stops training when the model's performance on the validation set stops improving.

Chapter 8

Results

This section explains the results acquired by performing the experiments mentioned in 6.

8.1 Analytical results

The exploratory analysis and time series decomposition reveal a seasonality on different levels, such as on a daily, weekly, and yearly level. In the day there are usually two peak hours of traffic, on average starting from 05.00 until 09.00, and the afternoon rush starting from 15.00 until 16.00. On the weekly basis, the traffic shows a rapid decrease in traffic during weekends. However, this decrease is not equally strong for all tunnels, but can actually maintain its level throughout the whole week.

8.2 Model predictions

This section shows the results acquired by running different experiments. The different experiments are using different set of tunnels and features so observe the change in performance and to gain a deeper understanding of the underlying patterns.

8.2.1 Experiment #1

The model is trained on data for all three urban tunnels. The results for this experiment can be found in Experiment1.txt.

Prediction on	**Baseline RMSE**	**Model RMSE**	**Baseline MAE**	**Model MAE**
Sub-urban tunnel 1	11.3	**16.1**	7.5	**12.1**
Sub-urban tunnel 2	10.5	**15.9**	6.9	**11.8**
Sub-urban tunnel 3	6.7	**12.1**	4.2	**8.7**
Urban tunnel 1	65.5	**41.1**	48.3	**31.1**
Urban tunnel 2	272.3	**112.3**	163.3	**69.5**
Urban tunnel 3	125.9	**60.6**	77.5	**41.5**

The model is performing better than baseline model based on the RMSE and MAE for the urban tunnels.

8.2.2 Experiment #2

Prediction on	**Baseline RMSE**	**Model RMSE**	**Baseline MAE**	**Model MAE**
Sub-urban tunnel 1	11.3	**17.9**	7.5	**11.9**
Sub-urban tunnel 2	10.6	**16.9**	6.9	**11.0**
Sub-urban tunnel 3	6.7	**15.4**	4.2	**9.1**
Urban tunnel 1	65.5	**39.1**	48.4	**28.5**
Urban tunnel 2	272.3	**106.9**	163.4	**65.2**
Urban tunnel 3	125.9	**59.1**	77.5	**39.0**

In this particular experiment no major change is detected. A small improvement is seen in RMSE for urban tunnels.

8.2.3 Experiment #3

Prediction on	**Baseline RMSE**	**Model RMSE**	**Baseline MAE**	**Model MAE**
Sub-urban tunnel 1	11.3	**15.0**	7.5	**11.0**
Sub-urban tunnel 2	10.6	**15.3**	6.9	**11.3**
Sub-urban tunnel 3	6.7	**14.2**	4.2	**10.0**
Urban tunnel 1	65.5	**38.5**	48.4	**28.7**
Urban tunnel 2	272.3	**104.0**	163.4	**62.9**
Urban tunnel 3	125.9	**58.0**	77.5	**39.0**

A small improvement is seen in the RMSE, but so significant. The performance compared to the baseline is very similar for all tunnels. However, by introducing sub-urban data into training set, the RMSE for sub-urban tunnel 2 which is Drengstig tunnel has improved from experiment 3. Furthermore, the RMSE for experiment for this tunnel remains the best so far based on the performed experiments.

8.2.4 Experiment #4

In this experiment the model is trained on all six tunnels and the category feature is added, and an evaluation of the model's performance can give an indication of the significance of this feature. This category feature is acquired through a spatial join between the location of points and tunnel. This is explained in section 4.3.

Prediction on	**Baseline RMSE**	**Model RMSE**	**Baseline MAE**	**Model MAE**
Sub-urban tunnel 1	11.3	**15.8**	7.5	**11.2**
Sub-urban tunnel 2	10.6	**15.2**	6.9	**11.7**
Sub-urban tunnel 3	6.7	**11.2**	4.2	**8.1**
Urban tunnel 1	65.5	**40.4**	48.4	**28.5**
Urban tunnel 2	272.3	**116.8**	163.3	**72.7**
Urban tunnel 3	125.9	**65.0**	77.5	**43.2**

The change in performance is negative and the values have started to increase for all of the tunnels.

8.2.5 Experiment #5

In this experiment the model is trained on all six tunnels and both the category and temporal features are added.

Prediction on	**Baseline RMSE**	**Model RMSE**	**Baseline MAE**	**Model MAE**
Sub-urban tunnel 1	11.3	**19.6**	7.5	**13.9**
Sub-urban tunnel 2	10.6	**17.8**	6.9	**12.8**
Sub-urban tunnel 3	6.7	**15.3**	4.2	**10.1**
Urban tunnel 1	65.5	**43.7**	48.4	**31.0**
Urban tunnel 2	272.3	**112.6**	163.4	**68.8**
Urban tunnel 3	125.9	**60.3**	77.5	**41.5**

It is worth mentioning that the subset of tunnels used in the experiments are not necessarily representative of the population, but rather a small sample. In 2022 there are 1260 tunnels in Norway [2], and this sample is very small subset of it. Therefore more data and elaborate testing may be performed to find conclusive remarks.

Chapter 9

Discussion

The experimental results are preliminary, and does not serve the basis for any clear conclusion. However, based on what is performed in this thesis, and the results acquired on the sample used, this section discusses different aspects of it.

Based on the results presented in section 8, and the experiments explained in section 6 the RNN model did perform better than the baseline for the urban tunnels. However, for the sub-urban tunnels the model was not able to show good results.

Q1: What are the trends and seasonal patterns observed in tunnel traffic data?

Based on the aggregated hourly traffic flow that is available publicly at the time of writing this thesis, the real-time traffic behaviour cannot be seen. However, based on the given data set the larger trends can be seen. The data analysis in section **??**, shows that the majority of tunnels follow a cyclical behaviour during the day, depicting the two rush hours from 05.00 to 09.00 and 15.00 to 16.00. This patterns is relatively recurring for all tunnels, while the magnitude of traffic flow and the amplitude of the cyclical fluctuations is what sets the tunnels apart. Tunnels like Auglend tunnel has a very high traffic flow during rush hours at 09.00 compared to the Drengstigtunnel at the same time. This is naturally due to their location. However, what is also revealed is that the sub-urban tunnels can have a higher traffic volume during weekends. This aligns with what can be expected as several of these tunnels are located near holiday destinations.

Q2: How does an RNN model trained on a single tunnel perform vs on multiple tunnels?

In experiment #2 the model is trained only on a single urban tunnel, the Storhaugtunnel, together with the temporal features. The goal was to evaluate if the patterns in one tunnel is enough to recognize the other, while also providing potentially helpful features. Compared to the previous experiment, the predictions on sub-urban tunnels got worse, while a small improvement in RMSE for the urban tunnels are seen.

Q3: Does adding the category feature (urban, or suburban) or temporal features improve the model performance?

The introduction of additional features such as the category and temporal features did not provide any significant performance gain. In experiment #2 when the model is trained on Storhaug tunnel together with the temporal features, the performance for urban tunnels are improved a little, but not significantly. The addition of category feature in experiment #4 did not improve the performance for any. Similarly, in experiment #6 both the temporal and category feature is added, providing the worst results compared to the other experiments.

What is interesting is that the results from experiment #2 did not provide results very different from the other experiments which includes more data. This can be related to that the urban tunnels posses a more typical or cyclical pattern seen in an urban tunnel like Storhaugtunnel. However, based on the experiments alone it can not be inferred if this feature is relevant or not. More tunnels can be introduced, as well as more traffic data. The literature review presented in the start of this work also indicated in some of the papers that additional features did not significantly improve the model.

9.1 Why is the model not performing well on sub-urban tunnels?

The fact that the model is not able to perform as good on sub-urban tunnels as it is on urban tunnels, does give an indication that there is a distinction of tunnels we are touching upon. There can be various reasons why it is unable to recognize the patterns in the sub-urban tunnels.

1. This can be related to that urban tunnels possess a more cyclical or predictive pattern based on the rush-hours. However, it is important to keep in mind that the data does not reveal the real-time behaviour of the traffic.

2. One of the major reason can be the need for more data. In this case the model is only seeing data worth of one year, where the first months are used for training, and the last months for testing. That means that the data the model is predicting on, it has never seen those patterns before. One idea is to add at least 1 more year of data to see if it is able to produce any significant changes in the performance. The other ways to tweak the performance can be to add more neurons, or more epochs or layers.

3. The choice of scaler could be a significant factor that can influence the model performance. The scaler used in this thesis, squeezed the tunnels into a 25th and 75th percentile which is standard for the RobustScaler. However, it may have been that the magnitude of variations between the tunnels may have got lost somewhere, as the model is not able to identify the sub-urban tunnels.

4. In this work, the population density is set to 200 in QGIS and this decides whether a tunnel is urban or not. This value was set based on visualization of the tunnels and points in QGIS, and also familiarity with the local area the tunnels are present in.

5. An idea is to investigate whether the urban and sub-urban traffic can be linked to the concepts of uninterrupted and interrupted traffic. Urban traffic tends to be more interrupted, influenced by traffic signals, congestion, and other urban factors.

However, there can be several reasons for why the model is not performing that well even after adding the cateogory feature. Firstly, the selected temporal features may not be as relevant for the sub-urban as they may be for urban. Features that quantify the amplitude during rush-hours, or other cyclical features, and also features such as the weekend should be used to evaluate further what works. As stated in both [9] and [10] the spatial and temporal features are suggested to be the most influential on the traffic. As stated in [9] the amount of data is also a big factor improving model accuracy, compared to using less data. Therefore other temporal features may be used, and more data must be utilized to be able to see the big picture.

9.2 Model performance

The RNN model is performing better than baseline for the urban tunnels. However, it is worth mentioning that the RNN is not a memoryless model as the baseline is. The RNN is given a sequence of 2 weeks, while the memoryless model only has the last value. The comparison can potentially reveal whether the traffic flow time series data possess temporal linearity or not. We can see from the results that for the sub-urban tunnels, the RNN did not outperform the baseline. Revisiting the Greenshield model explained in chapter 3.1.1 the uninterrupted traffic may possess linearity. Sub-urban traffic can be investigated further of the potential linearity and whether deep learning is the correct approach.

9.3 Why a simple RNN was selected?

The decision to employ a simple RNN model stemmed from the hypothesis that tunnel traffic exhibits less complexity compared to regular traffic, primarily due to the reduced occurrence of various disruptions such as animal crossings, traffic lights, or pedestrian activities. This does not mean that tunnel traffic patterns are not complex, but based on the literature review several advanced models were used for traffic forecasting due to its dynamic and non-linear nature. The more advanced a model is and the more parameters it has, the more extensive tuning and testing is required to fully understand its way or working.

9.4 Limitations

Acquiring relevant data proved to be the foremost challenge in this thesis. The majority of research paper data sets were not publicly available, necessitating the creation of a new data set, which became a significant challenge. Additionally, establishing the relationship between NVDB and Trafikkdata was another challenge. Despite both being owned by the Norwegian Public Road Administration, both data sources are maintained separately. This connection was crucial for linking the hourly traffic data to the corresponding tunnels. The connection to Trafikkdata's API was a necessary step to fetch the geographical coordinates of each traffic registration point, which was otherwise unavailable through the web interface.

A limitation of this thesis is the limited availability of data. Although several tunnels had nearby traffic registration points, not all of them had it. Furthermore, certain tunnels had intersections or other traffic deviations between them and the registration point, further limiting the dataset's scope. Consequently, the dataset used in this study represents a small subset due to these limitations. However, the installation of more registration points near tunnels by NPRA can gradually expand this dataset.

Time constraints also posed a limitation. To meet deadlines, certain tasks, such as file preprocessing, were performed using Excel.

Chapter 10

Conclusion

This work proposed a method for obtaining and integrating data from two different data sources NPRA to construct a data set of tunnel traffic. On this data set, a deep learning approach is proposed to predict the traffic flow for the next hour. The model is a simple RNN model that is able to handle single or multiple tunnels. Additionally, different experiments are run that tests the performance using temporal features and categorical features. Based on the sample used for testing, the addition of temporal features or categorical feature such as urban or sub-urban, did not improve the results drastically. The model performed better than baseline for urban tunnels, but was unable to capture the sub-urban patterns. This observation indicates that the catgory feature may be a valid distinction between them. However, as a feature fed into the model it does not provide significant changes. The results in this thesis are preliminary, and more extensive testing must be performed to reach any clear conclusive remarks.

10.1 Future work

The model is an simple RNN model that can be further tweaked or analyzed to detect any errors the model may have produced. The function that runs the model is created such that the user can insert parameters directly into it, and test with different combinations without having to deal with the back-end. Furthermore, other experiments can be performed to see the performance. Experiements testing with different lag features can be used, to see which lag features perform the best. This can indicate how far in history the temporal dependencies may exist. Secondly, a linear model such as an ARIMA model can be constructed for a comparison of performance. This can indicate whether the temporal dependencies between the traffic flow are linear or not. Additionally, the process can be automated by combining the output from the data fetch into the model, which requires more work. Finally, other machine learning models can be implemented for comparison of performance. This can indicate whether the deep learning approach is too complex for this tas. Also, the model can be run of a different data set to evaluate how well it is able to generalize. Finally, more values can be created in the category feature, such as semi-urban, or semi-sub-urban.

The github repository associated with this thesis can be found at `https://github.com/TunnelSafety/Tunnel-Traffic`.

,,

Bibliography

[1] E. L. Manibardo, I. Laña, and J. D. Ser, "Deep learning for road traffic forecasting: Does it make a difference?", *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6164–6188, 2022. DOI: `10.1109/TITS.2021.3083957`.

[2] Store norske leksikon, *tunnel*, Accessed: June 15, 2023. [Online]. Available: `https://snl.no/tunnel`.

[3] Statsforvalteren, *Ulykker på veg og i tunnel*, Accessed: June 15, 2023. [Online]. Available: `https://prosjekt.statsforvalteren.no/fylkesros-rogaland/risikoomrade/store-ulykker/ulykker-pa-veg-og-i-tunnel/`.

[4] Veier24, *Gudvangatunnelen sterkt skadet etter brann i vogntog - mandag ble det satt inn ferge — veier24.no*, Accessed: June 15, 2023. [Online]. Available: `https://www.veier24.no/artikler/gudvangatunnelen-mye-skadet-etter-trailerbrann-ma-stenges-i-lang-tid/461810`.

[5] P. Kanestrøm, "Traffic flow forecasting with deep learning", Available at `http://hdl.handle.net/11250/2563560`, M.S. thesis, Department of Computer Science, Norwegian University of Science and Technology, Trondheim, 2017.

[6] M. Levin and Y.-D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)", *Transportation Research Record*, no. 773, 1980.

[7] Y. Hou, P. Edara, and C. Sun, "Traffic flow forecasting for urban work zones", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1761–1770, 2015. DOI: `10.1109/TITS.2014.2371993`.

[8] Oregon State University, Portland State University, University of Idaho, *Types of Traffic Flow*, Accessed: June 2, 2023. [Online]. Available: `https://www.webpages.uidaho.edu/niatt_labmanual/chapters/trafficflowtheory/theoryandconcepts/TypesOfTrafficFlow.htm`.

[9] F. Schimbinschi, X. V. Nguyen, J. Bailey, C. Leckie, H. Vu, and R. Kotagiri, "Traffic forecasting in complex urban networks: Leveraging big data and machine learning", in *2015 IEEE International Conference on Big Data (Big Data)*, 2015, pp. 1019–1024. DOI: `10.1109/BigData.2015.7363854`.

[10] C. Hu, K. Xie, G. Song, and T. Wu, "Hybrid process neural network based on spatio-temporal similarities for short-term traffic flow prediction", in *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008, pp. 253–258. DOI: `10.1109/ITSC.2008.4732609`.

[11] F. Guo, J. W. Polak, and R. Krishnan, "Comparison of modelling approaches for short term traffic prediction under normal and abnormal conditions", in *13th International IEEE Conference on Intelligent Transportation Systems*, 2010, pp. 1209–1214. DOI: `10.1109/ITSC.2010.5625291`.

[12] M. J. Cassidy and R. L. Bertini, "Some traffic features at freeway bottlenecks", *Transportation Research Part B: Methodological*, vol. 33, no. 1, pp. 25–42, 1999. DOI: `https://doi.org/10.1016/S0191-2615(98)00023-X`.

[13] Statens Vegevesen, *Nasjonal vegdatabank API-dokumentasjon*, Accessed: June 2, 2023. [Online]. Available: `https://api.vegdata.no`.

[14] Digitaliseringsdirektoratet, *Norsk lisens for offentlige data (NLOD) 1.0*, Accessed: June 2, 2023. [Online]. Available: `https://data.norge.no/nlod/no/1.0`.

[15] Statens Vegvesen, *Vilkår for bruk av data vegvesen.no*, Accessed: June 2, 2023. [Online]. Available: `https://www.vegvesen.no/fag/teknologi/nasjonal-vegdatabank/hente-ut-og-se-pa-data-i-nasjonal-vegdatabank/vilkar-for-bruk-av-data/`.

[16] Statens Vegevesen, *Nasjonal vegdatabank*, Accessed: June 2, 2023. [Online]. Available: `https://www.vegvesen.no/fag/teknologi/nasjonal-vegdatabank/`.

[17] S. Vegvesen, *Trafikkdata API*, Accessed: June 14, 2023. [Online]. Available: `https://www.vegvesen.no/trafikkdata/api/`.

[18] Statens Vegvesen, *Om trafikkdata*, Accessed: June 2, 2023. [Online]. Available: https://trafikkdata.atlas.vegvesen.no/#/om-trafikkdata.

[19] Statens Vegvesen, *Trafikkdata - Om fartsdata*, Accessed: June 9, 2023. [Online]. Available: https://trafikkdata.atlas.vegvesen.no/#/om-trafikkdata#om-fartsdata.

[20] Statistisk Sentralbyrå, *About Statistics Norway*, Accessed: June 9, 2023. [Online]. Available: https://www.ssb.no/en/omssb/ssbs-virksomhet.

[21] Statistisk Sentralbyrå, *Kart SSB*, Accessed: June 9, 2023. [Online]. Available: https://kart.ssb.no/.

[22] Statens Vegvesen, *Nasjonal vegdatabank API*, Accessed: June 2, 2023. [Online]. Available: https://api.vegdata.no/.

[23] Jan Kristian Jensen, *Jobb interaktivt mot NVDB api V3*, Accessed: June 9, 2023. [Online]. Available: https://github.com/LtGlahn/nvdbapi-V3.

[24] Statens Vegvesen, *NVDB API Documentation*, Accessed: June 9, 2023. [Online]. Available: https://nvdbapiles-v3.atlas.vegvesen.no/.

[25] Statens Vegvesen, *Trafikkdata API*, Accessed: June 2, 2023. [Online]. Available: https://www.vegvesen.no/trafikkdata/api/.

[26] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: A deep learning approach", *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015. DOI: 10.1109/TITS.2014.2345663.

Aman Riaz, Spring 2023 – Masters in Data Science – University of Stavanger

Tunnel traffic prediction using machine learning

Can we use the potential in available tunnel traffic data to forecast congestion in tunnels?

Motivation & goal:

- Norway has over 1000 tunnels, some of which have a history of accidents and tunnel fires resulting in severe consequences.
- Statens Vegvesen owns traffic data with unrevealed potential.
- Leverage the potential of data using machine learning.
- Attempt to create a model that is generalized across different tunnels in Norway.

Main idea:

- Integrate traffic data from multiple open source databases owned by Statens Vegvesen into a unified dataset.
- Conduct spatial analysis to identify traffic registration points that are in close proximity to a tunnel and do not intersect with other roads beforehand.
- Utilize population data to determine the categorization of tunnels as urban or suburban.

Challenges:

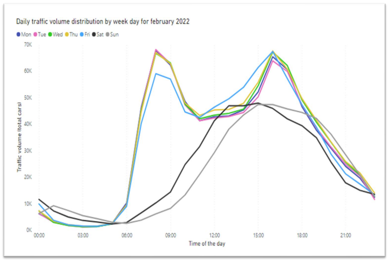
- The process involves comprehending the data and its interdependencies, and identifying approaches to integrate them into a cohesive dataset.
- Underwater tunnels in Norway present challenges for 2D data analysis.
- Develop a versatile model for predicting outcomes across various types of Norwegian tunnels.
- Discovering reliable data sources and constructing a comprehensive dataset from scratch has presented a notable challenge.
- Hourly aggregated data removes the detail or patterns in the data.

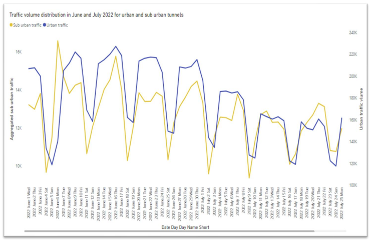
Findings:

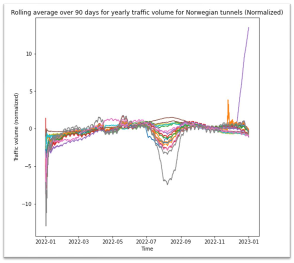
- Tunnel traffic is less complex than regular traffic as it lacks external factors like traffic lights and pedestrians.
- A big differentiating factor for tunnel traffic volume is whether the tunnel is categorized as urban or suburban.
- Tunnels exhibit consistent trends and seasonality throughout the year, with summer showcasing the highest variability across multiple tunnels (refer to graph on the right).
- Certain suburban tunnels in Norway experience higher traffic volumes during summers, mainly because they are situated along or in close proximity to popular holiday destinations.

Traffic data characteristics:

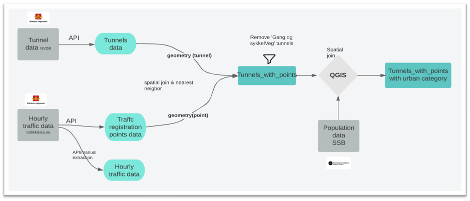
- Temporal: Temporal data is inherently time-dependent, necessitating the preservation of the order of data points.
- Spatial: traffic is dependent on location, and also to nearby locations
- Non-linear nature







Data collection:



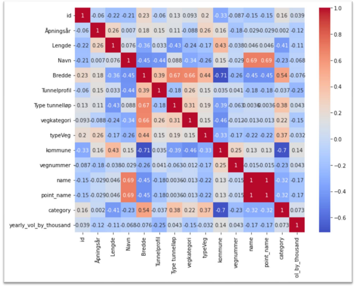
- Spatial join performed using geo dataframes, finding nearest traffic registration points within a specified distance, accounting for tunnels, resulting in a geo dataframe combining tunnel data with corresponding traffic registration names.
- Urban and suburban tunnels are identified by checking if their start and end points fall within selected 1 km x 1 km squares with a population above 200, and the results are written back to the file.
- The program initially attempts to fetch hourly data for multiple traffic registration points from an API, but due to server time-outs and occasional None responses, a manual approach is used to download the data from the trafikkdata.no API interface.
- The data from the file is separated into two dataframes: one for lane 1 and 3, and another for lane 2 and 4, ensuring that the traffic patterns of both directions are distinct and do not overlap.

Feature extraction:

- Extracted temporal features such as hour, month, dayofMonth
- The tunnels are categorized as urban or suburban based on their presence within a yellow box on the map, representing a grid where the total population in 2019 exceeded 200. If the whole tunnel is inside the grid/box the tunnel urban, or if one of the ends of the tunnel is inside the tunnel is categorized as sub-urban

Time series decomposition was conducted to partition the components of a time series into distinct parts, aiming to gain insights into their individual behavior.

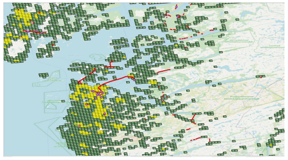
Correlation matrix

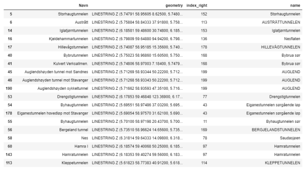


Data preprocessing:

- Data cleaning includes handling missing values and implementing dateTime indexing.
- To facilitate easier comparison and mitigate variations in traffic volume magnitudes across different tunnels, the data is scaled to a similar axis.
- The dataset is transformed into a windowed format to serve as input for the RNN model.

QGIS – tunnels, points and urban areas





	Navn	geometry	index_right	name
5	Storhaugtunnelen	LINESTRING Z (5.74791 58.95605 8.62500, 5.7480...	152	Storhaugtunnelen
6	Austrått	LINESTRING Z (5.75884 58.84333 37.91800, 5.758...	113	AUSTRÅTTUNNELEN
14	Iglatjørntunnelen	LINESTRING Z (6.18581 59.48600 30.74800, 6.185...	153	Iglatjørntunnelen
16	Kjeldehammartunnelen	LINESTRING Z (6.79809 59.64880 94.94200, 6.798...	136	Nesflaten
17	Hillevågstunnelen	LINESTRING Z (5.74087 58.95185 15.35600, 5.740...	178	HILLEVÅGTUNNELEN
40	Bybrutunnelen	LINESTRING Z (5.75023 58.96860 15.60500, 5.750...	168	Bybrua sør
41	Kulvert Verksallmen.	LINESTRING Z (5.74806 58.97003 7.18400, 5.7479...	168	Bybrua sør
45	Auglendshøyden tunnel mot Sandnes	LINESTRING Z (5.71269 58.93344 50.22200, 5.712...	199	AUGLEND
46	Auglendshøyden tunnel mot Stavanger	LINESTRING Z (5.71268 58.93343 50.22200, 5.712...	199	AUGLEND
190	Auglendshøyden sykkeltunnel	LINESTRING Z (5.71662 58.93593 47.35100, 5.716...	199	AUGLEND
53	Drengstigtunnelen	LINESTRING Z (6.17853 59.46848 123.36900, 6.17...	77	Drengstigtunnelen
54	Byhaugtunnelen	LINESTRING Z (5.69551 58.97486 37.03200, 5.695...	43	Eiganestunnelen sørgående løp
178	Eiganestunnelen hovedløp mot Stavanger	LINESTRING Z (5.69054 58.97570 31.62100, 5.690...	43	Eiganestunnelen sørgående løp
55	Byhaugtunnelen	LINESTRING Z (5.70100 58.97198 20.43700, 5.700...	11	Byhaugtunnelen sør
56	Bergeland tunnel	LINESTRING Z (5.73510 58.96624 14.65600, 5.735...	169	BERGJELANDSTUNNELEN
58	Nes	LINESTRING Z (6.31814 59.64333 14.09800, 6.318...	78	Saudasjøen
60	Hamra I	LINESTRING Z (6.18574 59.40068 50.25000, 6.185...	97	Hamratunnelen
143	Hamratunnelen	LINESTRING Z (6.18353 59.40274 59.56000, 6.183...	97	Hamratunnelen
113	Kleppetunnelen	LINESTRING Z (5.61823 58.77383 40.91200, 5.618...	114	KLEPPETUNNELEN

Model:

- RNN model preserves time-dependency in time series.
- SimpleRNN model in tensorflow performs well on single urban tunnel.
- Non-urban tunnel training captures seasonality, but struggles with cycle amplitude.