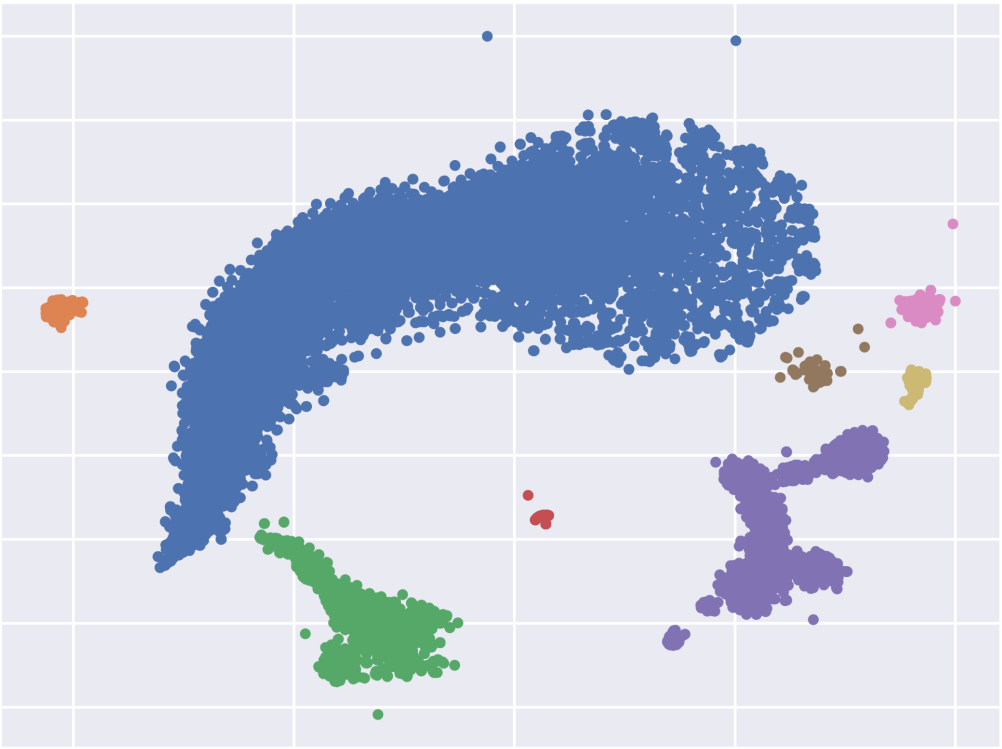
CHALMERS
UNIVERSITY OF TECHNOLOGY



Clustering and modeling of wireless backhaul data traffic

Can ML identify patterns in traffic data, and can we model urban traffic behavior in a simple way?

Master's thesis in Information and Communication Technology

JACOB BILLVÉN

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Clustering and modeling of wireless backhaul data traffic

Can ML identify patterns in traffic data, and can we model urban traffic behavior in a simple way?

JACOB BILLVÉN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Communication, Antennas and Optical Networks
Chalmers University of Technology
Gothenburg, Sweden 2025

Clustering and modeling of wireless backhaul data traffic
Can ML identify patterns in traffic data, and can we model urban traffic behavior in a simple way?
JACOB BILLVÉN

Supervisors: Rahul Devassy, Mikael Coldrey & Martin Sjödin, Ericsson AB
Zicong Jiang, Department of Electrical Engineering
Examiner: Giuseppe Durisi, Department of Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Communication, Antennas and Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: t-SNE embedding of 1344-dimensional traffic data.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

This thesis was partly funded by Sweden's Innovation Agency *Vinnova*.

Clustering and modeling of wireless backhaul data traffic
Can ML identify patterns in traffic data, and can we model urban traffic behavior in a simple way?
JACOB BILLVÉN
Department of Electrical Engineering
Chalmers University of Technology

Abstract

To handle the future demands of mobile broadband, knowledge of user behavior is of great importance. Knowing when and where user demand will be high allows for better planning of spectrum and computational resources. To aid this, this thesis investigates if the behavior of traffic throughput of wireless backhaul links from a European operator can be segmented into different clusters. We use both the k-means clustering algorithm and t-distributed stochastic neighbor embedding to attain the clusters. No distinct patterns emerge from the data, which instead appears to be uniformly spaced without clear boundaries.

To aid simulation of wireless links in future studies we also model the traffic behavior of urban links. The correlation between the model parameters as well as the error terms are calculated as a function of the geographical distance between the links. This helps decide whether links in proximity behave similarly or not. We find that the traffic on urban links has a similar shape but the model parameters are not correlated with respect to the distance between them. The parameters can be sampled from given distributions to generate synthetic traffic data.

Keywords: machine learning, wireless backhaul, traffic model, clustering, k-means

Acknowledgements

I would like to thank all colleagues at Ericsson Research who contributed to the enjoyable spring I spent in the Lindholmen office. Without you, the creation of this thesis would have been a lot more boring. To my main supervisor, Rahul Devassy, I present my greatest appreciation to all tips I've received; From how to navigate the slowness of large corporations to the mathematical definitions of things I only looked at as an engineer. To Mikael Coldrey and Martin Sjödin, I say thank you for the tips and ideas you gave when the original goal of this thesis fell short.

Finally, I would like to thank my fiancée Hanna for the support at home and for taking care of our puppy Arla. And the greatest thank you goes to Arla who has given me infinitely many lovely walks with never expiring energy, always carrying a stick.

Jacob Billvén, Gothenburg, May 2025

Glossary

Below is a short glossary of relevant terms in this thesis.

Traffic	Data rate over a link, e.g. 250 megabits per second
Traffic volume	Amount of data transmitted, e.g. one gigabyte
Clustering	Grouping similar items together. Similar to classification
Link	A bidirectional communication channel, here consisting of two wireless transceivers
Point	Used synonymously to a vector in $\mathbb{R}^n$

Nomenclature

Below is the nomenclature of acronyms, indices, variables, sets and functions that have been used throughout this thesis.

Acronyms

t-SNE	t-distributed stochastic neighbor embedding
Mbps	Megabits per second

Indices

i	Index of component in vector
t	Index for time step
n	Dimension of vector space

Variables

A, B	Amplitude parameters in traffic model [Mbps]
a, b	Example vectors in $\mathbb{R}^n$
y	Time series of traffic
α, β	Scaling parameters in $\mathbb{R}^+$
φ, θ	Angles
m	A large number in $\mathbb{R}^+$
ρ	Pearson correlation coefficient
Z, W	Example of random variable
μ	Mean value
σ	Standard deviation
k	Number of clusters in a run of k-means

Sets

C_I	Set of all points belonging to cluster I

Functions

z_t	Standardization function
d_E	Euclidean distance
S_C	Cosine similarity
d_C	Cosine distance

Contents

List of Figures

List of Tables

1

Introduction

Global mobile traffic volume is predicted to double in the five upcoming years [1]. To handle this demand, mobile network operators must provide reliable connectivity from their base stations to the internet. Fiber optic cables provide a reliable and high capacity link, 10 Gbps up to 120 km or 400 Gbps up to 40 km using plug and play solutions. However, installation costs can be high as digging the entire path between the nodes to lay the cable is required. A flexible alternative is to use wireless solutions to provide the link instead. These can operate over long distances and provide high capacity, 20 Gbps up to 100 km [2]. A base station far from any existing fiber infrastructure can establish a wireless link to a site already equipped with fiber, removing the need of expensive digging and thus reducing cost. Wireless backhaul can also be used to provide a cost efficient backup link to use if the fiber is unexpectedly out of order.

When planning a new link, wireless or wired, it is designed using estimates to handle the expected demands over the lifetime of the equipment. If the deployed capacity is too low the backhaul cannot fulfill the demands of its users and quality of service (QoS) decreases. Having a good prediction of user demand also enables better frequency allocation between links to reduce interference, further increasing QoS. We are therefore interested in identifying patterns in utilization of current links using machine learning methods for unlabeled data.

To facilitate research on wireless backhaul, realistic simulations of traffic is needed. The simulations can be used to develop algorithms for more efficient spectrum sharing or tuning of transmission power. The current way to simulate traffic is to consider the empirical cumulative probability distribution and randomly sample the traffic for each time slot from there. A model which also provides a realistic trend over the day, not just traffic with the correct statistical properties, would therefore increase accuracy of the simulations. We are therefore interested in creating such a model.

1.1 Research questions

The research questions this thesis answers are:

- Are wireless backhaul links clusterable based on their utilization data using machine learning algorithms k-means and t-SNE?
- What easily interpretable model can be made based on the data, and how would one choose parameters for this model? Are the parameters correlated as a function of link separation?

1.2 Scope and limitations

The utilization data used in this study was collected from an operator in Europe during 2022, from early July to late December. There are no guarantees that this data contains all wireless backhaul links in the operator's network. Also, as the backhaul traffic data for base stations served by fiber is unavailable, only a subset of all backhaul traffic data is analyzed. The temporal resolution of the data collection is 15 minutes which also limits the accuracy when determining concurrency of events.

There is a vast number of methods for clustering of time series, with Paparrizos, Yang and Li [3] listing 94 different methods from the last decade. It is therefore not possible to examine and evaluate them all in this thesis. The set of available algorithms is also reduced by Ericsson IT policies, restricting which external software libraries are allowed. Furthermore data is accessed and handled with Apache Spark [4] to enable parallel computing. Only a few clustering algorithms are implemented there, limiting the use of parallel processing. Some methods were instead run locally on only a subset of the data to not exceed reasonable computation times.

Creating a model for the general traffic pattern can be done in a multitude of ways. Our goal is to have a decent model that is easy to use and that has good interpretability. Therefore only models with few parameters are considered. To aid the choice of parameters, we also provide estimates of probability distributions to sample parameters from. The lack of accuracy due to the estimates of distributions will not be of great importance as many edge cases, such as short bursts of high intensity traffic, are already lost due to the sampling interval of 15 minutes.

1.3 Ethics and sustainability

The data used in this thesis does not contain any information that can identify individual users. Only group level statistics are available, removing concern for personal integrity violations. However, the data does contain information on the location of the wireless backhaul links. A malicious actor equipped with knowledge of utilization and location of each link will have a large advantage in destroying

critical communication services. The data was therefore kept in the Ericsson cloud without direct access to the internet to limit the risk of a leak.

Analyzing the dataset requires large compute resources due to its size ($\approx$ 100 million rows), and thus also requires electricity to function. The exact energy consumption is unknown, but it should be negligible compared to Ericsson's current usage. This reasoning also applies to network operators where the solution might be used in the future. The research could also help reducing power consumption in other devices lowering the total energy consumption in the long term.

2

Theory

Here we present the necessary background to the project. First we give a general description of wireless backhaul links, then the algorithms and methods used in the thesis are described.

2.1 Wireless backhaul networks

In the network architecture for a mobile operator the backhaul network connects a base station to the core network. The core provides high speed connectivity to the internet as well as operator specific servers managing the entire network. It is therefore vital to connect a base station to the core network to be able to provide communication services. The alternatives for this link are either fiber optic cables or wireless links, with fiber having a 60% market share in Western Europe but only 20% in India and Africa [5]. Figure 2.1 shows a setup where wireless backhaul provides an advantage over fiber, enabling connectivity over waterways and buildings without the need for expensive digging or laying cable. The wireless products in the Ericsson portfolio are capable of transmitting up to 25 Gbps and can manage link lengths up to 200 km. They span frequencies from 6 GHz to 80 GHz and are capable of using one or both polarizations.

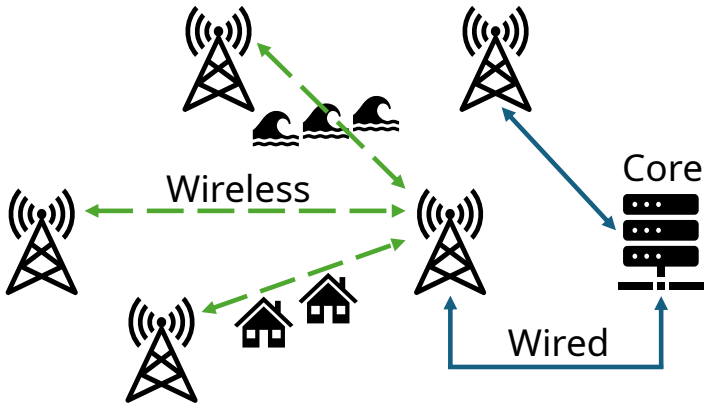


Figure 2.1: Graphic showing example deployment of both wireless and wired backhaul. Wireless solutions can ease installation over waterways or buildings where digging or laying cable might be troublesome or expensive.

2.2 Time series model

There are many models of time series, each fit for its specific setting. A simple model is the auto regressive one where $y_t = \alpha y_{t-1} + \beta y_{t-2} + ..., \ \alpha, \beta \in \mathbb{R}$, to ones where holidays and known special events are incorporated. These models provide a framework in which to analyze the properties of a time series, e.g. its trend, variance or periodicity.

In this thesis we will not consider time series as functions of its past values like

$$y_t = f(y_{t-1}, \ldots, y_0) \tag{2.1}$$

for value y at time t, but instead view them as functions of the current time

$$y_t = f(t). \tag{2.2}$$

This allows for representation of the series in a vector such that

$$y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix} \tag{2.3}$$

which puts the series in the vector space $\mathbb{R}^n$. Such construction allows for usage of standard vector operations instead of time series specific operations, such as Euclidean distance and clustering of ordinary points.

2.3 Euclidean and cosine distance

In time series analysis the similarity (or distance) between two series is often relevant to calculate. The most common choice is the Euclidean norm,

$$\text{Euclidean distance} = d_E(a, b) = ||a - b||_2 = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{2.4}$$

for two time series (or vectors by our notation) a and b. Another possibility is to use the cosine similarity

$$\text{cosine similarity} = S_C(a, b) := \cos(\theta) = \frac{\sum_{i=1}^{n} a_i b_i}{||a||_2 \cdot ||b||_2} \tag{2.5}$$

which effectively measures the angle θ between two vectors a and b. Note that vectors are similar when θ is small and S_C large. To transform the similarity to a distance measure where small values imply closeness it is customary [6] to use

$$d_C(a, b) := 1 - S_C(a, b). \tag{2.6}$$

The difference between Euclidean and cosine distance is best illustrated with an example: using $a = [5, 10, 15, 20]$ and $b = 10 \cdot a = [50, 100, 150, 200]$ then the Euclidean distance is large but the cosine distance small. The cosine similarity therefore provides better comparison of time series with the same shapes but different amplitudes compared with the Euclidean one [7].

2.4 K-means clustering algorithm

The k-means algorithm is a self-supervised machine learning method to cluster similar points into groups. As opposed to classification, where the classes are known a priori, clustering tries to figure out which classes there are. The algorithm works as follows [8]:

1. Select k, the number of clusters you want to group your data into
2. Select starting points for each group
3. Assign the data points to the cluster center they are closest to
4. Update the cluster center to the mean of its data points
5. If the cluster center moved, start again from 3. to refine its position

Step 2 can be done in a multitude of ways. The implementation used in this work is *scalable k-means++* (also named *k-means||*) by Bahmani et al. [9], which allows for fast processing over large datasets. In step 3, the definition of closest is dependent on the choice of distance metric.

2.4.1 Standardization of distance metrics in k-means

For both cosine similarity and Euclidean distance the input time series has to be standardized by computing their z-score. For a sample y_t its z-score is calculated as

$$z(y)_t = \frac{y_t - \mu_t}{\sigma_t} \tag{2.7}$$

where μ_t is the mean of all samples at time t and σ_t the standard deviation of the same samples.

In the case of Euclidean metric for k-means, standardization improves performance over using the raw data [10]. To see why, assume a fixed j and two vectors a, b where all components $a_i, b_i \in [0, 1]$ except $a_j, b_j \in [m, 2m] \ : \ |m| \gg 1$. As $||a - b||_2^2 = \sum_i (a_i - b_i)^2$, most terms in the sum will be in $[0, 1]$ except when $i = j$, where the term will be in $[0, m^2]$. And as $m \gg 1$ that term will dominate the result. By instead rescaling to zero mean and unit variance all components influence the result equally. The scaling is intuitively equivalent to $a_i \in [-1, 1] \ \forall \ i$ where all components obviously carry the same weight.

To see why standardization improves performance in the case of cosine similarity, we assume two vectors a, b where all components $a_i, b_i \in [0, 1]$ except $a_j, b_j \in [m, m + 1] \ : \ |m| \gg 1$ for a fixed j. Then we have

$$S_C(a, b) = \frac{\sum\limits_i a_i b_i}{\sqrt{\sum\limits_i a_i^2} \cdot \sqrt{\sum\limits_i b_i^2}} \approx \frac{m^2}{\sqrt{m^2}\sqrt{m^2}} \approx 1 \tag{2.8}$$

and thus the components $i \neq j$ do not affect the result. The same is true for variables where the range is large, e.g. if $a_j, b_j \in [-m, m] \ : \ m \gg 1$ then there will be occurrences when that component dominates the result. It is therefore of importance to standardize all variables to make sure they have equal influence to the distance metric.

2.5 Silhouette and silhouette score

The silhouette of a point, not to be confused with the silhouette score, is a tool to measure how well data has been clustered. It was presented by Rousseeuw in 1987 and the original paper [11] has a comprehensive description of it. Kaufman and Rousseeuw then expanded the concept to a silhouette score, being the average silhouette of a dataset [12]. The score ranges from -1 to 1 where a higher value is better, and it should have its maximum when computed over a clustering which used the same number of clusters as naturally exist in the data. When deciding which k to use in the k-means algorithm one should therefore choose the one with the highest silhouette score [13].

In general the score relies on two concepts, *similarity* and *dissimilarity*. Similarity measures the average distance between the vector a and all other vectors in the same cluster:

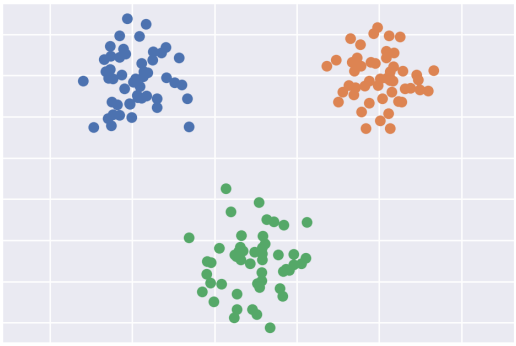
$$\text{sim}(a) = \frac{1}{|C_I| - 1} \sum_{b \in C_I, a \neq b} d(a, b) \tag{2.9}$$

where $a \in C_I$ with C_I being the set of all vectors assigned to cluster I, and $d(a, b)$ the distance between the two points a and b.

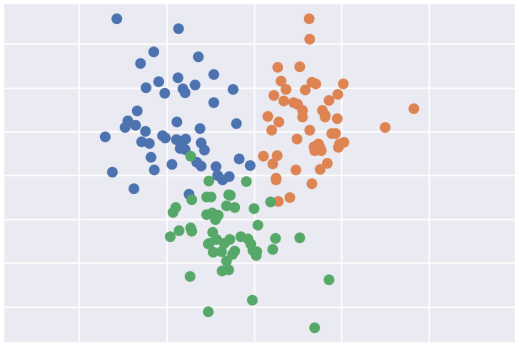
Dissimilarity instead measures the distance from a vector to the nearest *other* cluster:

$$\text{dis}(a) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{b \in C_J} d(a, b). \tag{2.10}$$

These values are then calculated for each and every vector in the dataset. If $\text{sim}(a) \gg \text{dis}(a)$ then the cluster is tight and far from the closest other cluster. If $\text{sim}(a) \approx \text{dis}(a)$ the point a is as far from its own cluster as from its closest neighboring cluster. If $\text{sim}(a) \ll \text{dis}(a)$ then the vector is closer to another cluster than it is to its own, and the vector is probably mislabeled. An example of data with different similarities can be seen in Figure 2.2.



(a) Data with clear clusters. Each cluster has high similarity and low dissimilarity. The silhouette score is 0.78.



(b) Data with no clear clusters. Each assigned cluster has low similarity and high dissimilarity. The silhouette score is .44.

Figure 2.2: Example of data with different similarities and dissimilarities.

These concepts are then used to compute the silhouette for the point a

$$s(a) = \begin{cases} 1 - \text{sim}(a)/\text{dis}(a), & \text{if sim}(a) < \text{dis}(a) \\ 0, & \text{if sim}(a) = \text{dis}(a) \\ \text{dis}(a)/\text{sim}(a) - 1, & \text{if sim}(a) > \text{dis}(a) \end{cases} \tag{2.11}$$

and the silhouette coefficient as

$$\tilde{s} = \frac{1}{|C|} \sum_{a \in C} s(a) \in [-1, 1] \tag{2.12}$$

where higher values are better.

2.6 t-distributed stochastic neighbor embedding

The t-distributed stochastic neighbor embedding (t-SNE) algorithm is a dimensionality reduction method that attempts to preserve significant structures of the original space in a low-dimensional space. The reduction is usually to a two or three-dimensional space to enable visual interpretation for humans. Maaten and Hinton describe the method comprehensively in [14] where they introduced the use of student's t-distribution in the SNE algorithm. The general idea is to preserve the probability that two points are neighbors in both the high and low dimensional spaces. The probability is given from the t-distribution in one dimension (distance between points) and minimization is done over the Kullback-Leibler divergence of the probabilities in the high and low-dimensional spaces. When calculating distance, both Euclidean and cosine distances are valid options.

The t-SNE algorithm has a parameter called *perplexity*, which affects how points are repelled and attracted from each other during the minimization. *Intuitively* it

truncates the student's t-distribution, setting it to zero above a certain distance and thus considering all points above the threshold equally distant. A higher value results in higher repulsion between clusters and thus greater separation between the clusters in the final space. The exact value thus has an effect on the clarity of the resulting representation, and one should try a range of values to find a good fit for the specific application. However, depending on the value of perplexity, the algorithm may introduce artifacts or group random noise into distinct clusters. Wattenberg, Viégas and Johnson show this in [15].

Importantly, the t-SNE algorithm creates a set of vectors that is similar to the high dimensional set of vectors with respect to the neighbors of each point. The result is only a bijection between the sets, not the spaces. After a run of the algorithm for one set of high dimensional vectors there is no way to determine where an additional unseen high dimensional vector would have its corresponding point in the low dimensional space. Thus it is not possible to train a t-SNE model on some data and then use that model for new data.

2.7 Pearson correlation coefficient

The Pearson correlation coefficient [16], given as

$$\rho_{Z,W} = \frac{\operatorname{cov}(Z,W)}{\sigma_Z \sigma_W} \in [-1,1] \tag{2.13}$$

for two random variables Z, W and their standard deviations σ_Z, σ_W, provides a normalized measure of correlation. A value of 1 indicates perfect correlation, a value of -1 perfect anticorrelation and a value of 0 no correlation.

2.8 Symmetric mean absolute percentage error

Symmetric mean absolute percentage error (SMAPE) is a measure which gives a percentage-like relative difference between two numbers. It is defined as

$$\operatorname{SMAPE}(a,b) = \frac{1}{n}\sum_{i=1}^{n}\frac{|a_i - b_i|}{|a_i| + |b_i|} \tag{2.14}$$

for two vectors a, b. The symmetric property makes $\operatorname{SMAPE}(a,b) = \operatorname{SMAPE}(b,a)$, which is beneficial when comparing pairs of vectors in a collection without regard of order.

3

Methods

Below, the methods used in the thesis are described. The available data together with the preprocessing steps is presented first, followed by the specific steps taken when clustering the data or fitting the model to it.

3.1 Data description and preprocessing

The data was collected from a European operator from July to December 2022. It contains location and utilization data for each link. The utilization data, which measures how many bits per second are sent over the link compared to its maximum capacity, is reported in a format like *in the last 15 minutes, how many seconds were spent in x to $x + 5\%$ utilization* where $x \mod 5 = 0$, thus resulting in 20 bins. The utilization is reported based on the received traffic, i.e. link A-B reports how much A received from B. Every entry also contains an identifier for which unidirectional link the data refers to and the max capacity for that link. An example is shown in Table 3.1 below.

Link ID	Timestamp	Max speed	0 to 5%	...	95 to 100%
A-B	2022-09-01 13:00:00	1050 Mbps	43 s	...	0 s
B-A	2022-09-01 13:00:00	890 Mbps	10 s	...	6 s

Table 3.1: Example of raw data of link utilization.

In total there are 97 million entries of the above kind. Only 0.005% of the entries differ by more than 5 seconds from the expected sampling time of 900 seconds (15 minutes), e.g. the utilization bins were filled with data for only 10 minutes due to a system shutdown or 20 minutes due to inability to upload the data to the collection server. Missing entries were filled with linearly interpolated values. All records with a sampling time from 7 to 31 minutes were included to not interpolate when data was available.

The histogram-like data was then reduced to *average traffic* and *standard deviation* from collected interval, all measured in megabits per second. The center of the bin was used to calculate the average. For each link, each statistic was assembled to a

vector of length one day, see Table 3.2 for an example, and one of length one week. Vectors with more than 10% interpolated values were discarded. The final dataset contained 5,842 unidirectional links, 100,755 vectors of entire weeks and 806,004 vectors of entire days.

Link ID	Date	Traffic [Mbps]	Standard dev. [Mbps]
A-B	2022-09-02	$[44.21, \ldots, 50.39]$	$[5.44, \ldots, 2.97]$
B-A	2022-09-02	$[31.68, \ldots, 22.75]$	$[7.63, \ldots, 3.72]$

Table 3.2: Example of link statistics assembled into vectors. Traffic and standard deviation are measured in Mbps.

Links were considered urban if either end of the link lay in the residential area of a city. Only cities with a population above $300,000$ were used, with no city having a population above $1,000,000$. Only 56 out of the six thousand unidirectional links were located in such areas. Further, only the direction with highest traffic volume was used, i.e., if link A-B used 25 Gb and B-A used 10 Gb of data during a day then B-A was discarded. The decision on which direction to use was made based on the data of Thursday 6th of October, a date chosen to not coincide with any weekends, holidays or school breaks and in the middle of the collection interval.

3.2 Clustering

The clustering was done using both k-means and t-SNE. The methods differ and are described below.

3.2.1 K-means

To perform k-means clustering, the vectors containing average traffic and standard deviation for one week were concatenated. By also using standard deviation, links with varying utilization could be differentiated from those with constant throughput. The concatenated vectors were then normalized in each component to zero mean and unit variance as described in section 2.4.1. The k-means algorithm was then run on the dataset, sweeping over a range of possible number of clusters, k, as well as using both Euclidean and cosine distance. For each k the silhouette score was computed. Both k-means and silhouette score was implemented by the Python library PySpark which allows for fast parallel computation. Note that methods of cluster validation other than silhouette score exist, but due to good performance [17] and available library implementation in Apache Spark [4] silhouette scoring will be the only method used in this work.

3.2.2 t-SNE

The t-SNE algorithm was run on vectors of length one day, considering only Thursdays in October, to reduce the computational load. The vectors were assembled by

concatenating the existing vectors containing average traffic and standard deviation of one day for each link. To reduce the computational load, only Thursdays in October were included. The vectors were then standardized and fed into the t-SNE algorithm in Python library Scikit Learn. The perplexity parameter of the run was sweeped to see which gave the most intuitive visualization for both Euclidean and cosine distance. The clusters were then identified manually and characteristics of each cluster was manually inspected. The inspection aimed to identify how the traffic data in the clusters differed, e.g. if there was more traffic in the morning for one cluster and more traffic in the night for another cluster.

3.3 Traffic modeling

Using the vectors containing one day of data for urban links, a sum of sinusoids with parameters A_i, B, φ_i

$$\begin{aligned} &B + \sum_{i \in \mathbb{Z}^+} A_i \cdot \sin\left(\frac{2\pi}{24} it + \varphi_i\right) \\ &A_i, B \in \mathbb{R}^+, \varphi_i \in [0, 2\pi) \end{aligned} \tag{3.1}$$

was fitted to the data using non-linear least squares. Note that time is measured in hours. By using sinusoids with frequencies of multiples of 24 hours, consistency over midnight was guaranteed. As real data is continuous, no drastic changes should appear at midnight. A positive A_i is enforced by modifying the phase. The optimal number of terms was tested, but due to interest in a simple and interpretable model only one sinusoid was used for the analysis. The parameters of the fitted model were then analyzed for correlation between links as a function of the distance between them, the link separation distance. Only links in the same city were compared with each other. If A is less than 5 Mbps, we only consider the constant part B, as the collected data is almost flat and thus the phase is indeterminable. In total, 6 links had A below 5 Mbps and 22 above out of the 28 considered urban.

A and B were compared using SMAPE for each pair of links each day, yielding an average SMAPE for each pair. For example, Link1 and Link2 generated one term $\frac{|A_1|-|A_2|}{|A_1|+|A_2|}$ for each day they both had data, and the average of those terms was then used in the analysis. In total there were 111,708 pairs. The phase, φ, was compared for each link pair by calculating the absolute phase difference $\Delta\varphi \in [0, 180°]$, defined as $\Delta\varphi := \min[|\varphi_1 - \varphi_2|,\ \ 360° - |\varphi_1 - \varphi_2|]$ due to the periodicity of the phase.

Parameters of probability density functions were obtained by fitting the function to the distributions of the empirically obtained model parameters using non-linear least squares. These were then used to give a general characterization of the distribution of the parameters, which could be used to create realistic models when needing synthetic data. The residuals, considered noise, were analyzed with the Pearson coefficient of correlation as a function of link separation distance. Also here, parameters of probability density functions were obtained by fitting the function to the noise distributions.

4

Results and Discussion

Here we present the findings of the work, together with discussions regarding relevance, accuracy and future studies. First, the clustering is described for k-means and t-SNE, followed by the modeling of traffic. The modeling consists of three parts, choice of model complexity, the distribution if its parameters when fitted to the data and finally the characterization of the noise.

4.1 Clustering

The results and discussion for clustering is split to cover each method, k-means and t-SNE, separately. t-SNE provides better separation of different traffic patterns, although it only separated flatlines of different amplitudes from more sinusoidally shaped patterns.

4.1.1 K-means

To evaluate the performance of the k-means algorithm, we plot the silhouette score versus number of clusters, see Figure 4.1. The best score is attained for two clusters and decreases for an increasing number of clusters. While the indication to select two clusters as the probable k is strong, manual inspection of the clusters attained from both cosine and Euclidean distance show that they consist of series either with some daily variation or those that constantly measure a flat line of $0-5\%$ utilization. Due to different max capabilities, the $0-5\%$ flatlines occurred at multiple Mbps values.

Removing flatlines from the dataset has no major impact, as seen in Figure 4.1. Choosing two clusters is still the best option, and by inspection these show to describe two lines similar to the two upper ones in Figure 4.2. Deeper examination on a map show no clear patterns for where the flatline links were located compared to those with variation. Proximity to a town has no visible impact, except that very high traffic links are located in large cities.

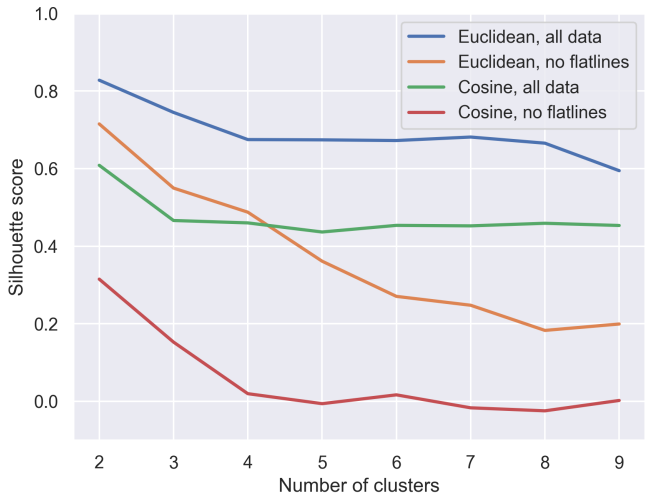


Figure 4.1: Graph over the silhouette score for different number of clusters for both distance metrics. A higher score is better.

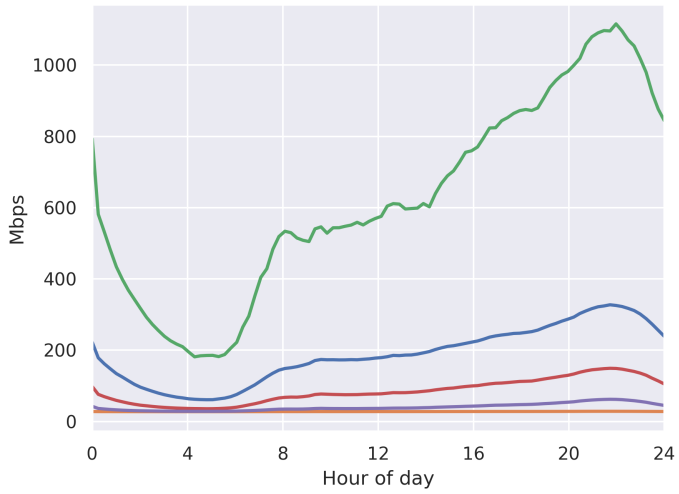


Figure 4.2: Clusters found for average 15 minute traffic on Thursdays by k-means with five clusters and Euclidean distance. All lines have a similar trend over the day.

Further, the cluster centers found for an increasing number of clusters show the same shape but with different scaling, see Figure 4.2. This, together with decreasing silhouette scores indicate that there is no clear boundary between the different clusters. Instead we believe that the traffic data for most links can be written as

$$\text{traffic} \approx \Gamma \cdot \alpha + \beta \cdot \mathbb{1} \tag{4.1}$$

where Γ is the vector containing the general trend with low traffic during night and increasing traffic during the day, $\alpha, \beta \in \mathbb{R}^+$ are scaling parameters and $\mathbb{1}$ is the all ones vector. As the scaling is continuous, no boundary can be drawn to separate the data. The assumption of no clear boundary is strengthened by the fact that traffic volume is a function of the number of users served by each base station [18]. It is reasonable to assume that there exist base stations with number of users ranging from very low to very high, and everything in between.

4.1.2 t-SNE

The t-SNE representations show clear clusters, with Euclidean distance seen in Figure 4.3 and cosine similarity in Figure 4.4. By manual inspection the blue cluster in both figures shows to contain links with a major daily variation. The brown cluster for cosine distance shows no difference to the blue one. All other clusters in both figures contain versions of approximately flat lines of different amplitude as described above. Visualizing the links on a geographical map show no other trends, such as flatlines only occurring in rural areas, with regard to cluster belonging.

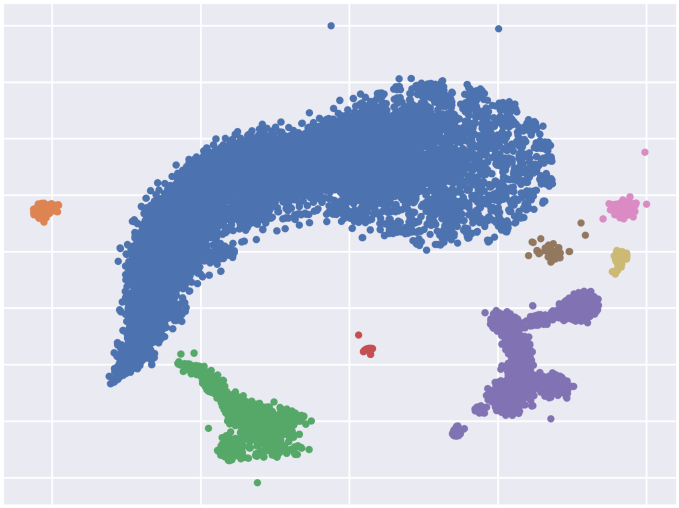


Figure 4.3: t-SNE (perplexity= 50) representation with Euclidean distance and manually colored clusters.

The blue central clusters in both figures strengthen the idea of a continuum of traffic patterns as described in (4.1). While the shape of the cluster produced by t-SNE only partially reflects the shape of the actual data, the fact that all links with any daily variation are placed in one continuous cluster indicates that the traffic patterns in the original dataset also lack a separation boundary.

4.1.3 General discussion on clustering

The estimate of traffic model given in (4.1) is a broad generalization for the traffic behavior. The continuum produced by the assumption is what we believe makes the problem hard to solve with the methods used here. However, one could define a set of rules for how to classify the traffic data, e.g., see if traffic volumes are higher on

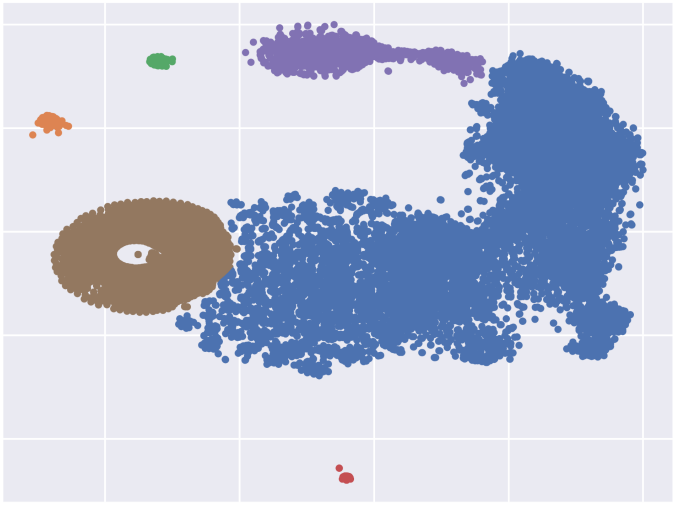


Figure 4.4: t-SNE (perplexity= 100) representation with cosine distance.

weekends or weekdays. This would no longer be a self-supervised machine learning algorithm as someone has to specify these rules, and thus it falls outside the scope of this thesis. We do however suggest more research into the different patterns seen close to e.g., malls, commuting roads, factories etc ..., from which one could assemble a few scenarios with peak traffic appearing on weekday mornings, weekday evenings, weekend evenings etc. These predefined scenarios could then be used to segment the data.

The continuum seen in the t-SNE representation would be interesting to investigate further. Points lying on opposite ends of the continuum should, as neighbors lie close to each other, be different in shape. We have made no analysis of the changing shapes inside a cluster, but by looking into these one could perhaps attain some useful representations. One could also choose a set of uniformly distributed points on the boundary of the cluster and investigate whether these show a meaningful changes between them. The geographical belonging of the links in different part of the cluster could also carry some meaning.

4.2 Traffic modeling

The traffic modeling will be described in three parts: first, how to select the number of sinusoidal terms in the model, then how the parameters of the selected model are distributed, and last how the noise behaves.

4.2.1 Choice of model complexity

To investigate the optimal number of sinusoidal terms in the sum that constitutes the model, we plot the average Pearson correlation coefficient of the model noise in Figure 4.5. The correlation is calculated for each link pair for each day they both have data and then averaged over all such pairs. Only pairs with both links in

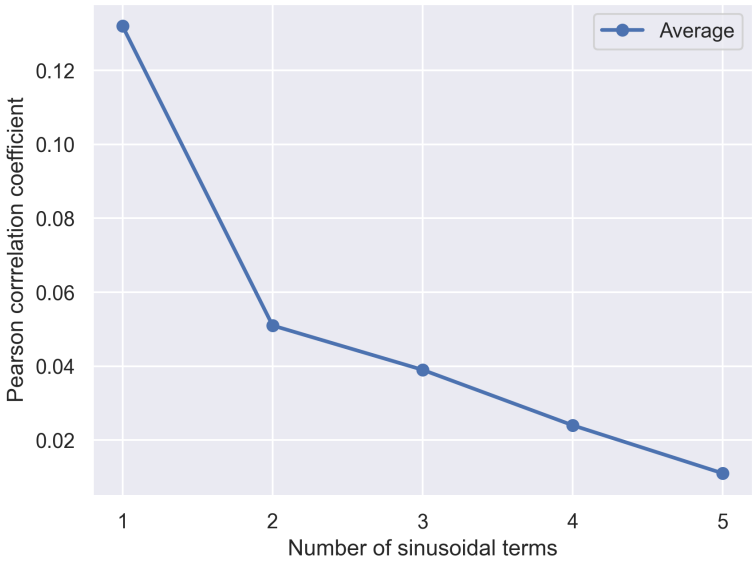


Figure 4.5: Average Pearson correlation coefficient of the noise for different number of sinusoidal terms. The correlation is calculated for link pairs in the same city and averaged over all pairs and days.

the same city are considered. The gain of using two sinusoidal components is large compared to using just one, as seen by the decreasing correlation of the noise. A lot of the noise correlation in the one sinusoidal case is rather the model not being able to correctly represent the traffic pattern. Figure 4.6 provides an example of such, where the green curve better fits the collected data around 12 o'clock. Using three or more sinusoidal terms reduces correlation but at a lower rate than before, indicating that two terms is a reasonable tradeoff between accuracy and number of parameters.

However, to ease interpretability we use only one sinusoidal term:

$$\text{Traffic model}(t, A, B, \varphi) = A \cdot \sin\left(\frac{2\pi}{24}t + \varphi\right) + B + \text{noise} \tag{4.2}$$

with free parameters $A, B \in \mathbb{R}^+$, $\varphi \in [0, 2\pi)$. Note that t is measured in hours and A, B in Mbps for convenience. The noise is the residuals of the model with respect to the data.

4.2.2 Parameters of the simple model

We fit the above model to all time series of urban links, resulting in one set of model parameters per link per day. In Figure 4.7 the SMAPE of parameters A and B is shown. No trend is seen over link separation distance.

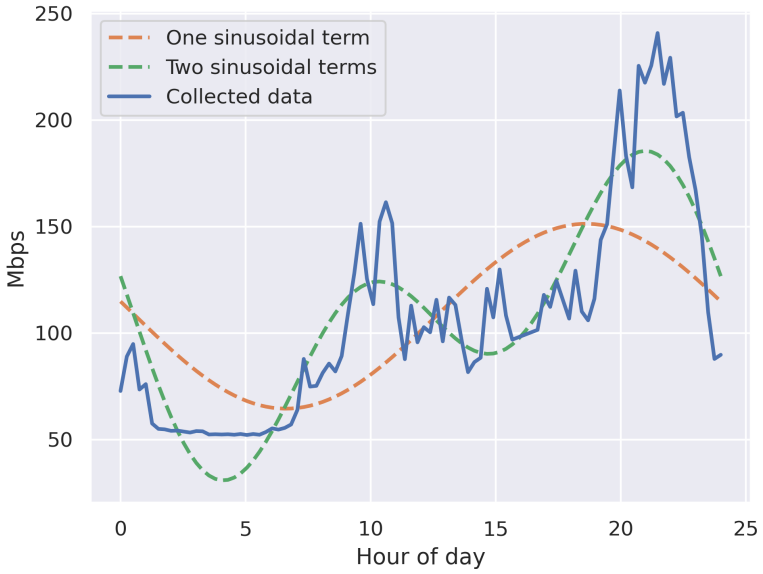
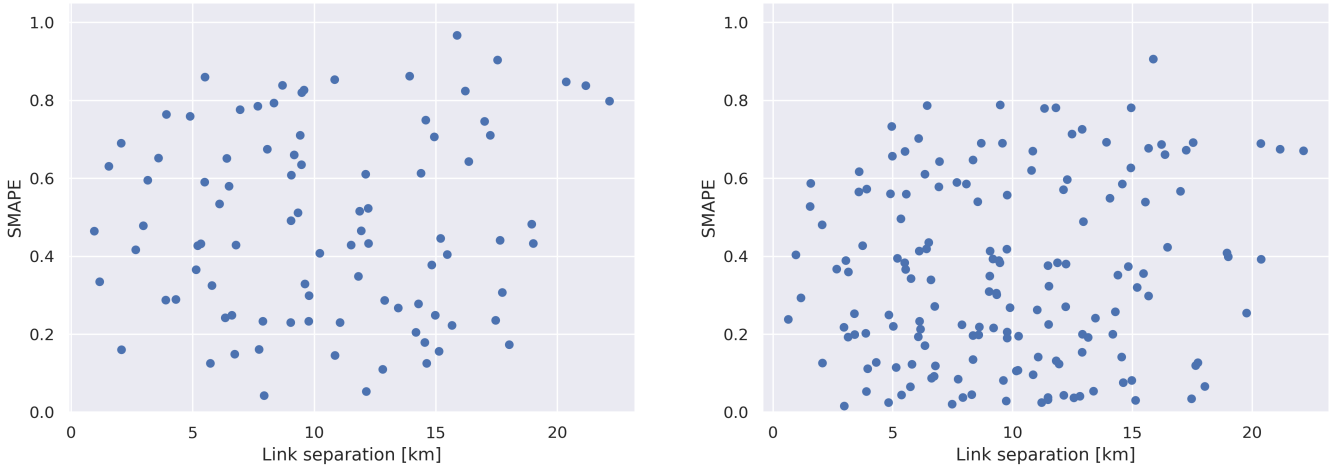


Figure 4.6: Example comparison between traffic models with one and two sinusoidal terms, and the collected data. The two term model fits the data better around 12 and 21 o'clock.



(a) Average SMAPE of parameter A.

(b) Average SMAPE of parameter B.

Figure 4.7: SMAPE of model parameters A and B in $A \cdot \sin(\frac{2\pi}{24}t + \varphi) + B$ as a function of link separation distance. A small SMAPE indicates that values are close. No trend is seen over distance.

For the absolute phase difference $\Delta\varphi$, illustrated in Figure 4.8, we observe differences ranging from 15° to 130° with no major trend over distance. All differences above 80° appear from 7 to 15 km of separation. This indicates that links with separation distance in the interval 7 to 15 km have a larger phase difference compared to links

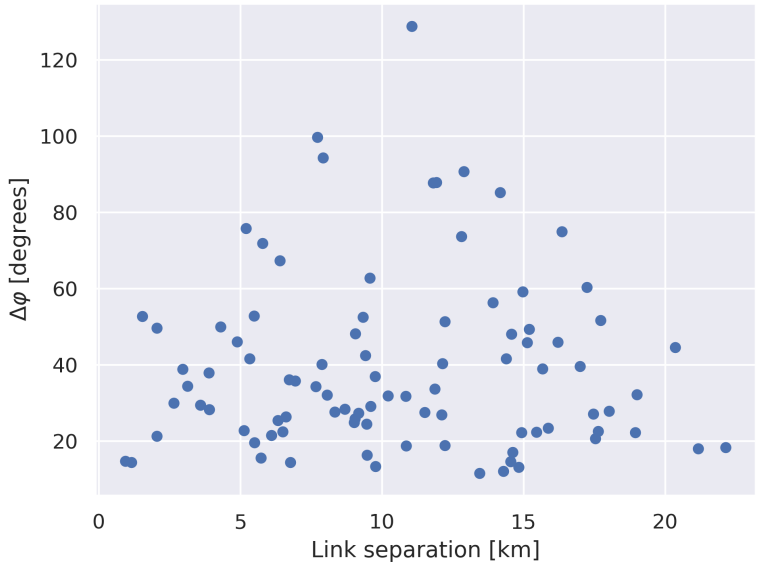
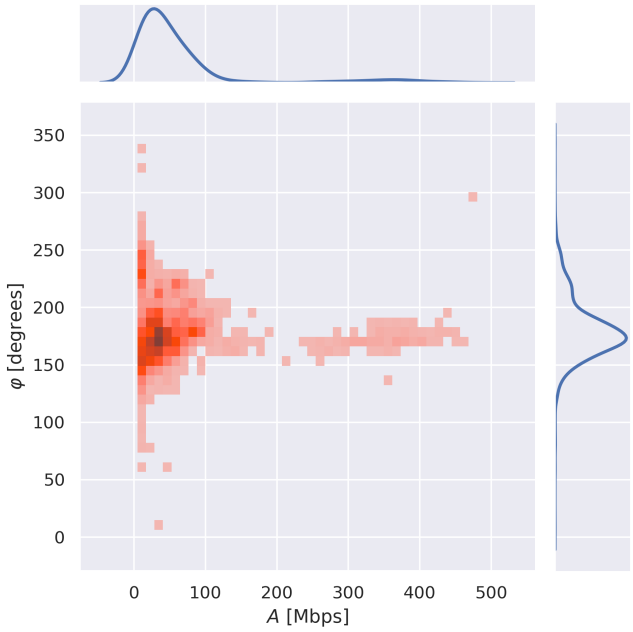


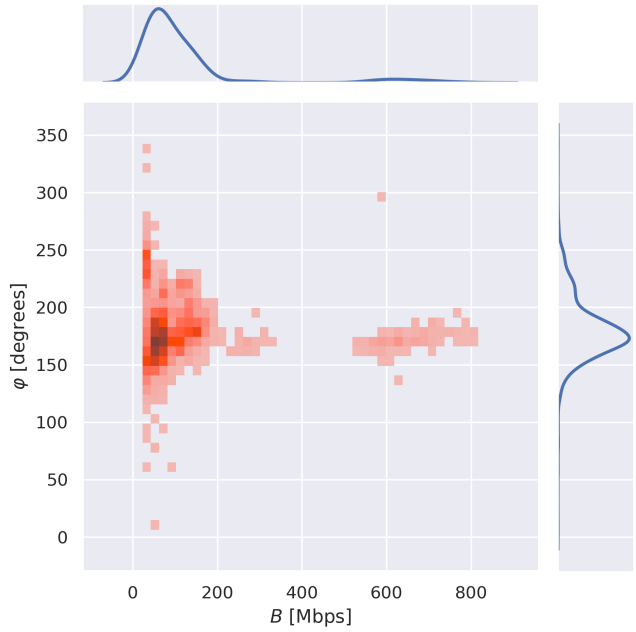
Figure 4.8: Average absolute phase difference $\Delta\varphi$ as function of link separation. No trend is seen as a function of link separation distance.

outside it. However, the separation interval also contains many links with $\Delta\varphi < 80°$, and no strong conclusion can be drawn. Also, as most $\Delta\varphi$ are smaller than 60°, most links approximately have a valley in the early morning and a peak in the evening, see Figure 4.6 for an example. A difference of 180° would indicate the opposite, a peak in the morning and a valley in the afternoon.

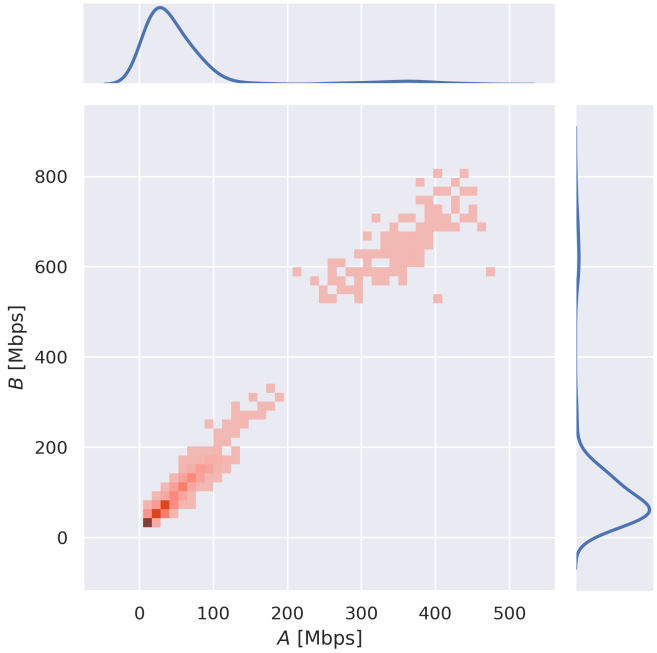
The joint distributions of all parameters when $A > 5\,\text{Mbps}$ are shown in Figure 4.9. Most links have $A \approx 40\,\text{Mbps}$, $\varphi \approx 175°$ and $B \approx 60\,\text{Mbps}$. The phase $\varphi \approx 175°$ corresponds to a peak at about 18:00. One link has $A \approx 350\,\text{Mbps}$ and $B \approx 650\,\text{Mbps}$. This link is probably an aggregate one, serving multiple sites. It thus sees a higher traffic demand as it carries the sum of all site it serves.



(a) Distribution of A and φ.



(b) Distribution of B and φ.



(c) Distribution of A and B

Figure 4.9: Distribution of model parameters A, B and φ in $A \cdot \sin(\frac{2\pi}{24}t + \varphi) + B$ for all links where $A > 5\,\text{Mbps}$. Every link appears once for every day it has a fitted model. The blue curves show the (one dimensional) kernel density estimates of the parameter.

The Pearson correlation matrix of A, B, φ is shown in Table 4.1. The correlation between A and B is large, but between φ and A or B the correlation is small. This shows that φ is uncorrelated with both A and B.

	A	B	φ
A	1.000	0.990	-0.066
B	0.990	1.000	-0.070
φ	-0.066	-0.070	1.000

Table 4.1: Correlation matrix of A, B and φ.

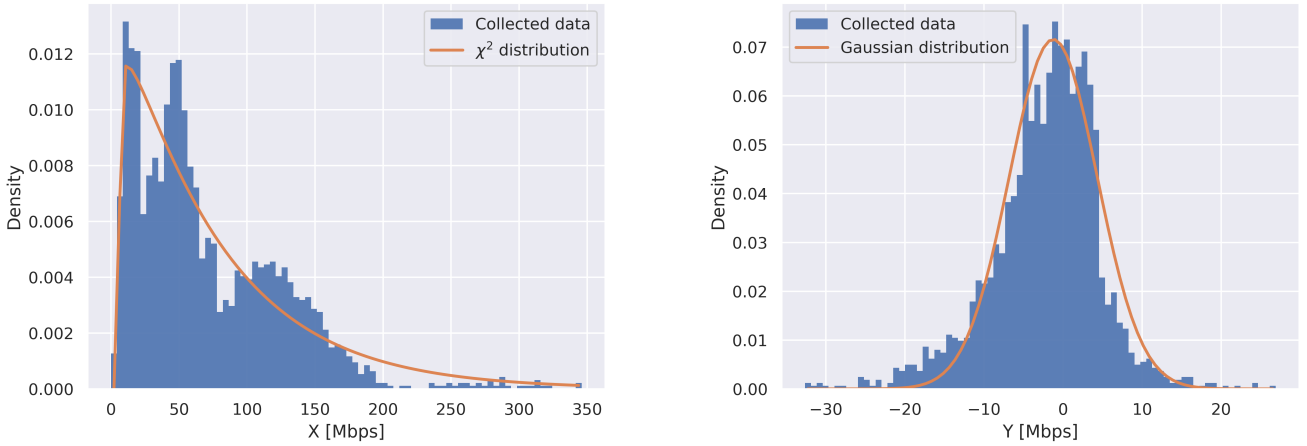
In Figure 4.9c we see that the samples approximately lie on a straight line. To be able to draw samples from a distribution corresponding to this, we transform each pair $[A, B]^T$, constraining $5 < A < 200$ to avoid the single link appearing where $A \approx 350$ and $B \approx 650$, as

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \left(\begin{bmatrix} A \\ B \end{bmatrix} - \begin{bmatrix} \min A \\ \min B \end{bmatrix} \right) \tag{4.3}$$

and thus

$$\begin{aligned} A &= \min A + X\cos(\theta) - Y\sin(\theta) \\ B &= \min B + X\sin(\theta) + Y\cos(\theta) \end{aligned} \tag{4.4}$$

where the minimization is done over all A values, dito for B, and the matrix is the *clockwise* rotational matrix in two dimensions with $\theta = \text{avg}\left[\arctan\left(\frac{B-\min B}{A-\min A}\right)\right]$. The values from this dataset are $\theta = 60.9°$, $\min A = 5.0\,\text{Mbps}$ and $\min B = 22.9\,\text{Mbps}$. This aligns the line seen in Figure 4.9c with the x-axis. With these new variables, we get a Pearson correlation of -0.330 between X and Y. For simplicity, we consider this low enough to assume lack of correlation between the random variables. We then fit a χ^2 distribution to X and a Gaussian distribution to Y, see Figure 4.10.



(a) Actual data and fitted distribution of X.

(b) Actual data and fitted distribution of X.

Figure 4.10: Data and fitted distributions for X and Y.

The Gaussian for Y has a mean of -1.2 Mbps and a standard deviation of 5.6 Mbps. The χ^2 distribution for X has 2.1 degrees of freedom*, a location of 6.5 Mbps and scale of 34.3 Mbps. Location shifts the PDF sideways and the scale adjusts the scale of the x-axis. This is comparable to standardization, where location is the mean and scale the standard deviation. See [19] for implementation details. The distributions were chosen to have a similar shape to the data. Especially, as X is nonnegative it calls for a nonnegative distribution, which χ^2 is.

To find the distribution of φ we fit a χ^2 distribution to the data†, see Figure 4.11. The distribution has 13.0 degrees of freedom, a location of 125.9° and a scale of 4.2°.

This method to draw parameter values for the traffic model is very coarse, and only gives an estimate of the actual traffic. However, it is simple and enables simulations with realistically shaped traffic patterns. A richer model, e.g., with two sinusoidal terms, would better capture the traffic behavior but at the cost of more parameters to set. We therefore believe that the simplicity of the sampling recipe given above outweighs the lack of accuracy.

*While the original definition is based on degrees of freedom being an integer, the function describing the PDF places no such constraint.

†The PDF must be scaled appropriately so that the integral from 0 to 360° is equal to 1.

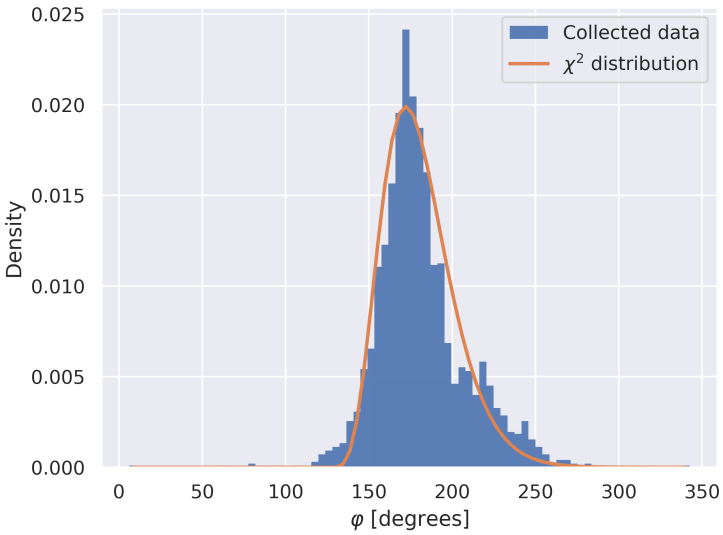


Figure 4.11: Data and fitted distribution for φ.

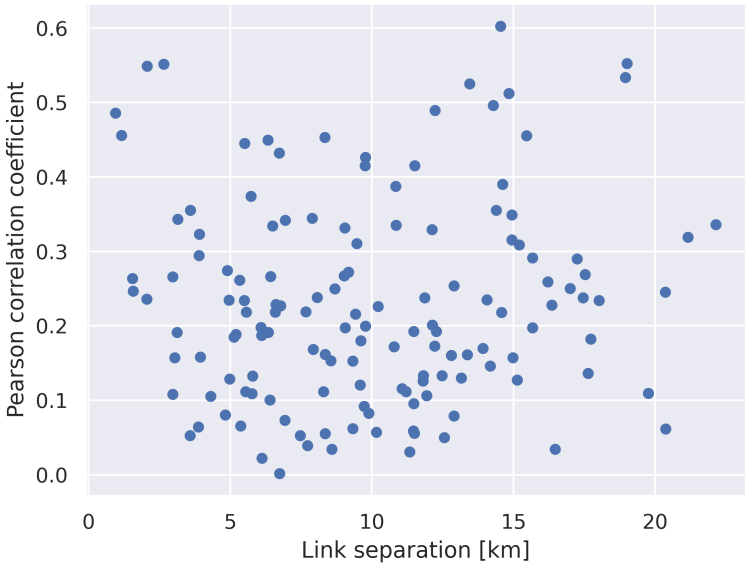
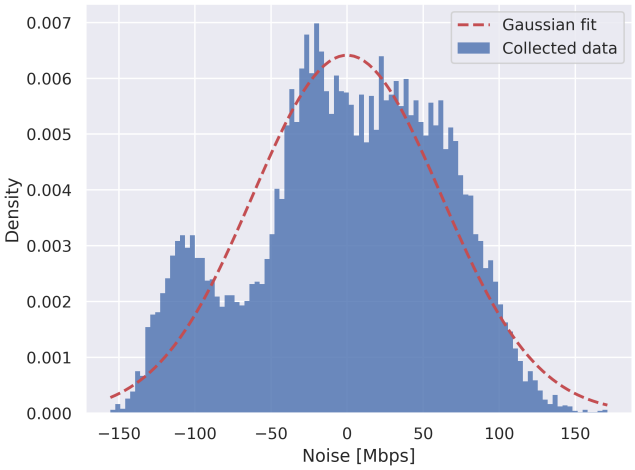


Figure 4.12: Average correlation of noise in the fitted models. Note that all correlations are positive and that no trend is seen over distance. The average is taken over each link pair.

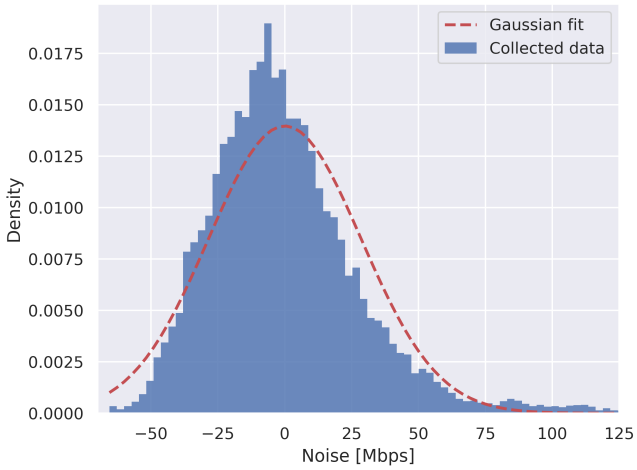
4.2.3 Characterization of model noise

The noise in the model, defined as $noise = collected\ data - model$ has its Pearson correlation shown in Figure 4.12. All correlations are positive, indicating that the links vary in similar ways. However, no trend is observed over distance. Positive correlation is expected as the behavior of a link is a function of the users it serves and for most base stations the users behave rather similar. The average correlation is 0.22, indicating a generally weak relationship.

Two examples of distributions of noise are shown in Figure 4.13. Note that the figure contains all noise for all daily models over the total data collection period for two different links. The left figure does not resemble a Gaussian, whereas the right one is considerably closer. In general, the Gaussian looking shape is the most prevalent. However, the Shapiro-Wilk test rejects the hypothesis that the data is drawn from a normal distribution for *all* links with 99% confidence. Although not statistically proven, for the sake of this analysis we consider the distributions to be Gaussian with the observed mean and standard deviation. We do this as the resemblance between the empirical data and the Gaussian probability density function, seen in Figure 4.13, is rather high. For all observed noises the means were less than 0.2 Mbps, and we thus assume the mean to be exactly zero.



(a) Distribution of noise for a link.



(b) Distribution of noise for a link.

Figure 4.13: Distributions of noise for two different links. A Gaussian is drawn on top with the same mean and standard deviation as the collected noise data. The Shapiro-Wilk test rejects the hypothesis that the data is drawn from a normal distribution for both links with 99% confidence.

To determine the standard deviation σ of the Gaussian noise, we examine the quantity σ^2/B, see Figure 4.14. The noise variance is expected to scale with the traffic volume, and normalizing with B scales all variances to the same range. The plotted χ^2 distribution has 2.0 degrees of freedom, a location of 1.3 Mbps and a scale of 6.1 Mbps. With B sampled with the previously presented method, the Gaussian noise's variance can be randomly and independently drawn from the presented χ^2 distribution. When sampling the noise to add to the sinusoidal model, the constraint that the measured traffic is positive must be honored by setting negative values to zero.

4.2.4 General discussion on modeling

No trend over distance is seen in any parameter. This is probably caused by the fact that geographical distance between links does not affect the type of area the base station is serving. An urban base station is approximately equally probable to have a factory or mall in a given distance, even if malls are mostly active on weekends and the factory during normal work hours, thus correlating very differently with the urban one. To find some correlation, we believe that the time resolution of the data must be increased to see if traffic bursts lasting a couple of seconds are correlated. Also, the amount of links is quite small with just 28 in urban settings. If instead the traffic from all sites was analyzed, including those connected via fiber, some correlations could perhaps be seen as that dataset would include plenty more base stations with smaller separation than the current dataset.

The proposed model also assumes constant noise variance over the day. It is rea-

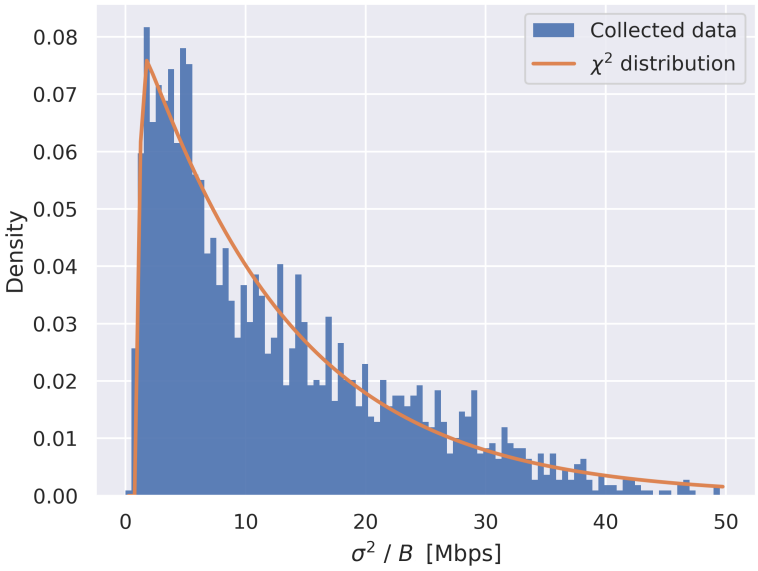


Figure 4.14: Distribution of σ^2/B and a fitted χ^2 distribution.

sonable to assume that variance would be greater when many users are active, i.e., during the day, and low during the night when most are asleep. This would, however, make the model harder to understand and use. There are many ways to improve general performance of the model, as seen above for correlation of noise for an increasing number of sinusoidal terms. We do suggest that further research should be done to find a better yet simple model.

Rotating the trend seen between A and B to attain X and Y requires the line to pass through the origin. This is done in this thesis by subtracting the minimum values from each component. A more robust way would be to fit a straight line to the distribution and then define X as the direction of that line and Y the orthogonal direction.

Noise also has some correlation between links, as seen in Figure 4.12. This should be taken into consideration when generating the noise for multiple links. This has not been considered in this work, but introducing some covariance in the Gaussian noise is a reasonable extension to this model. One could use the mean correlation of 0.22 to attain the noise covariances given the individual variances, but 0.22 indicates less correlation than the correlation of X and Y, which is -0.33. It is therefore contradictory to consider 0.22 a significant correlation, but not -0.33.

5

Conclusions

From the clustering part of this thesis, we conclude that no useful clustering appears using neither k-means nor t-SNE. While this doesn't imply that no patterns exist, the methods used here are insufficient to extract them. Other methods might give useful results, such as incorporating comparison between peaks of different days, and we encourage more research into that.

From the traffic model part, we conclude that no correlation as a function of link separation distance is seen for any model parameter. Using a model where link separation distance does not matter, we propose a method for how to generate realistic sinusoidal data using random samples from provided distributions to attain model parameters. While it has its limitations, it provides a basis for future work.

Bibliography

[1] "5G will carry 80 percent of mobile data traffic globally in 2030," Telefonaktiebolaget LM Ericsson AB, Tech. Rep., Nov. 2024, p. 9. [Online]. Available: https://www.ericsson.com/4adb7e/assets/local/reports-papers/mobility-report/documents/2024/ericsson-mobility-report-november-2024.pdf.

[2] *Ericsson and Telstra achieve world-first high speed capacity link to King Island*, Jul. 2024. [Online]. Available: https://www.ericsson.com/en/press-releases/7/2024/ericsson-and-telstra-achieve-world-first-high-speed-capacity-link-to-king-island (visited on 05/13/2025).

[3] J. Paparrizos, F. Yang, and H. Li, *Bridging the Gap: A Decade Review of Time-Series Clustering Methods*, Version Number: 1, 2024. DOI: 10.48550/ARXIV.2412.20582. [Online]. Available: https://arxiv.org/abs/2412.20582 (visited on 04/04/2025).

[4] *Apache Spark*. [Online]. Available: https://spark.apache.org/.

[5] "Backhaul media for 5G and beyond," Tech. Rep., Oct. 2023. [Online]. Available: https://www.ericsson.com/4a7ed9/assets/local/reports-papers/microwave-outlook/2023/backhaul-media-for-5g-and-beyond.pdf.

[6] Wolfram Research, *CosineDistance*, 2007. [Online]. Available: https://reference.wolfram.com/language/ref/CosineDistance.html (visited on 04/02/2025).

[7] A. Singhal, "Modern Information Retrieval: A Brief Overview," *IEEE Data Eng. Bull.*, vol. 24, pp. 35–43, 2001.

[8] E. Kavlakoglu and V. Winland, *What is k-means clustering?* Jun. 2024. [Online]. Available: https://www.ibm.com/think/topics/k-means-clustering (visited on 04/10/2025).

[9] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," en, *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, Mar. 2012, ISSN: 2150-8097. DOI: 10.14778/2180912.2180915. [Online]. Available: https://dl.acm.org/doi/10.14778/2180912.2180915 (visited on 04/10/2025).

[10] C. Wongoutong, "The impact of neglecting feature scaling in k-means clustering," en, *PLOS ONE*, vol. 19, no. 12, N. Aunsri, Ed., e0310839, Dec. 2024, ISSN: 1932-6203. DOI: 10.1371/journal.pone.0310839. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0310839 (visited on 04/10/2025).

[11] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," en, *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, Nov. 1987, ISSN: 03770427. DOI: 10.1016/0377-0427(87)90125-7. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/0377042787901257 (visited on 04/22/2025).

[12] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis* (Wiley Series in Probability and Statistics), en, 1st ed. Wiley, Mar. 1990, ISBN: 978-0-471-87876-6. DOI: 10.1002/9780470316801. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9780470316801 (visited on 04/22/2025).

[13] D. M. Saputra, D. Saputra, and L. D. Oswari, "Effect of Distance Metrics in Determining K-Value in K-Means Clustering Using Elbow and Silhouette Method," en, in *Proceedings of the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)*, Palembang, Indonesia: Atlantis Press, 2020, ISBN: 978-94-6252-963-2. DOI: 10.2991/aisr.k.200424.051. [Online]. Available: https://www.atlantis-press.com/article/125939938 (visited on 04/24/2025).

[14] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[15] M. Wattenberg, F. Viégas, and I. Johnson, "How to Use t-SNE Effectively," *Distill*, 2016. DOI: 10.23915/distill.00002. [Online]. Available: http://distill.pub/2016/misread-tsne.

[16] E. Weisstein, *Statistical Correlation*. [Online]. Available: https://mathworld.wolfram.com/StatisticalCorrelation.html (visited on 04/25/2025).

[17] E. Schubert, "Stop using the elbow criterion for k-means and how to choose the number of clusters instead," en, *ACM SIGKDD Explorations Newsletter*, vol. 25, no. 1, pp. 36–42, Jun. 2023, ISSN: 1931-0145, 1931-0153. DOI: 10.1145/3606274.3606278. [Online]. Available: https://dl.acm.org/doi/10.1145/3606274.3606278 (visited on 04/24/2025).

[18] "Exploring how traffic patterns drive network evolution," Telefonaktiebolaget LM Ericsson AB, Tech. Rep., Jun. 2023, pp. 23–25. [Online]. Available: https://www.ericsson.com/49dd9d/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-june-2023.pdf (visited on 04/22/2025).

[19] *Scipy.stats.chi2*. [Online]. Available: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.chi2.html#scipy-stats-chi2 (visited on 05/06/2025).

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY