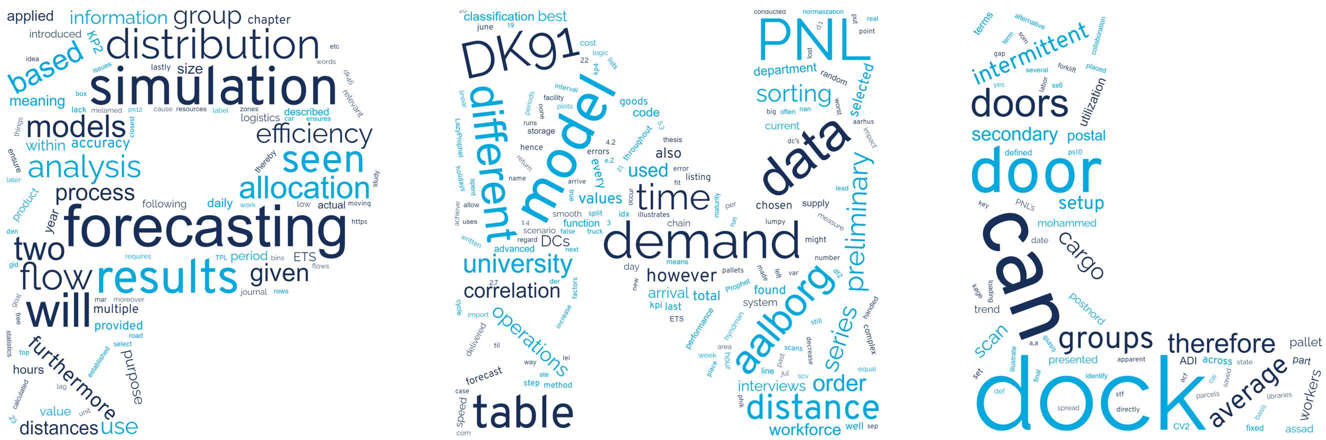
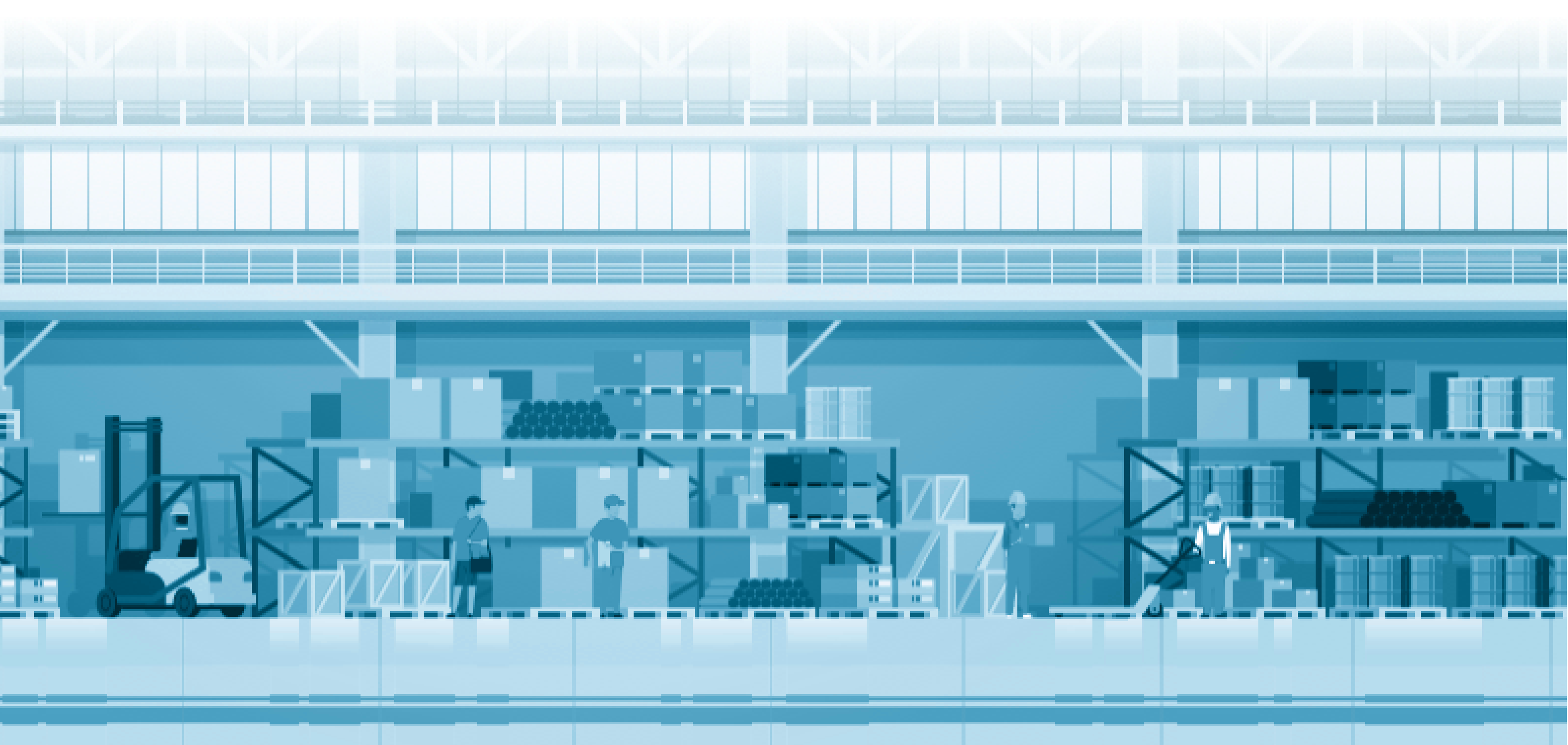
MASTER THESIS

FLOW OPTIMIZATION IN DISTRIBUTION CENTERS

VS4 OPERATIONS & SUPPLY CHAIN MANAGEMENT





Faculty of Engineering and Science
4th Semester - Management Engineering - Operations and Supply Chain Management
Fibigerstræde 16 - 9220 Aalborg Øst

AALBORG UNIVERSITY
MASTER THESIS

Project Title:

Flow Optimization in Distribution Centers through Simulation and Forecasting

Project:

P10 Master Thesis

Project Period:

February 2023 - June 2023

Project Group:

VS4 PostNord Project Group

Participants:

Carl-Emil Houmøller Pedersen
Markus Germann Knudsen
Victor Bartholomæussen

Supervisor:

Sven Vestergaard
Hans-Henrik Hvolby

Pages: 97
Words: 28087
Appendixes Pages: 21
Submission Date:
May 31st, 2023

Synopsis:

This project was developed in collaboration with Postnord Logistics. Through an initial analysis, it was found that Distribution Center (DC) costs are a high cost factor in the daily operations of PNL. Therefore, the pre-analysis examined the different DCs and their performance. It was found that Aalborg was one of the most efficient DCs measured on PNL's Scan KPI, however, also exhibited fluctuating demand. Furthermore, it was tested if demand correlated with efficiency on the KPI measure. An ensuing analysis of the current DC operations found that the dock door allocation was outdated. Therefore the facility was mapped and a mathematical model was developed to estimate the potential results. This provided a reduction between 3.9 and 4.3 hours per week. Moreover, this logic was implemented in a simulation model to better represent reality in terms of different parameters such as work schedules and changes in the velocity of the electric pallet lifters, etc. These results indicated that approximately the distance could be reduced by 10.8% and a 3.5% reduction in time. However for the results to be applicable in real life, real-time data was needed on a daily basis. Therefore it was chosen to forecast the demand on specific demand groups and use this as input instead. This was further elaborated upon through a solution proposal.

Resume

Dette projekt er skrevet i samarbejde med PostNord Logistics (PNL) som er en del af koncernen Posten AB. Posten AB er splittet i tre forgreninger, hvoraf PNL udgør den ene. Gennem en initierende analyse bliver organisationen belyst fra forskellige vinkler ift. finansielle resultater, omkostningsfordeling og operationelle funktioner. Ud fra disse, samt interviews og et tidligere kenskab til virksomheden, blev det tydelig gjort at der ikke var blevet kigget på interne processer i de forskellige terminaler som PNL opererede.

Herfra blev der lavet en række initierende analyser i forhold til størrelsen på terminalerne, deres demand samt individuelle flow. Ydermere blev der foretaget korrelationsanalyser på mængden af demand og effiktiviteten af de enkelte terminaler, for at se om der kunne belyses en sammenhæng. Ud fra forskellige faktorer såsom; interviews, geografisk placering og tid, blev det besluttet at terminalen i Aalborg skulle være udgangspunktet for analysen. Ud fra dette samt forskellige kost-faktorer såsom labor-cost og transportation cost, blev der formuleret en problemstilling der skulle svare på om en effektivisering af de nuværende processer i Aalborg terminalen var muligt. Gennem analysen blev de forskellige flows og processer i terminalen yderligere analyseret. Dette blev først gjort igennem en matematisk analyse, hvorefter resultaterne blev eftervist i et simuleringsprogram for bedre at repræsentere virkeligheden og dets parametre. Ud fra både den matematiske modellering og simuleringen, var der tydeligt at løsningen ville bidrage til en øget effektivitet hos Aalborg terminalen. Hertil blev det belyst at PNL ikke har den nødvendige datamodenhed til at applikere en sådan løsning i praksis. Derfor blev der udført forecasting, for at skabe en midlertidig løsning på problemet. Ud fra forskellige forecasting metoder kunne det konluderes, at forecasten ikke nødvendigvis gav de bedste resultater afhængigt af hvilke data-grupper man belyste. Trods resultaterne, blev det stadig vurderet at alternativet var bedre end ikke at ændre i de nuværende processer, hvortil forecasting kunne være en løsning. Slutteligt er et løsningsforslags afsnit udarbejdet med henblik på at benytte de fundene resultater i praksis. Dette inkludede konkrete forslag til hvordan man kan ændre processflowet og derved reducere tid og distancer. Ydermere blev det tydeliggjort at der vil være andre fordele som en øget scanrate indgående varer.

Abbreviations

- BU → Business Unit (Term)
- CEP → Courier, Express & Parcel (Industry)
- C&E → Courier & Express (BU)
- PDK → Post Danmark A/S (BU)
- PN → PostNord Group AB (Organisation)
- PNL → PostNord Logistics A/S (BU)
- P&G → Pallets and Groupage (BU)
- RFO → Road Freight Operations (BU)
- TPL → Third-Part Logistics (Term/BU)
- Distribution Center → DC
- SS → Secondary Sorting (Process)
- PS → Preliminary Sorting (Process)
- ETS → Error, Trend, Seasonality (Forecasting Method)
- ARIMA → Autoregressive Integrated Moving Average (Forecasting Method)
- NNAR → Neural Network Autoregression (Forecasting Method)
- Prophet → Facebook Prophet (Forecasting Method)
- LightGBM → Light Gradient Boosting Machine (Forecasting Method)
- MAE → Mean Average Error (Performance Measure)
- MAPE → Mean Absolute Percentage Error (Performance Measure)
- MASE → Mean Absolute Scaled Error (Performance Measure)
- ADI → Average Demand Interval (Demand Classification)
- CV → Coefficient of Variation (Demand Classification)

Participants:

The signature of each participant confirms equal terms for the process of development of the project. Hence, each person is responsible for all contents of this paper.

Carl-Emil Houmøller Pedersen | Markus Germann Knudsen | Victor Bartholomæussen

Contents

Introduction 1

1.1 Company introduction

The purpose of this chapter is to introduce the company of interest for the project. The section includes a description of the organizational structure and general order flow, to establish a foundation to examine the organization.

1.1.1 Organisation

The project will revolve around Postnord Logistics (PNL), part of PostNord Group AB (PN). To understand the operational value of PNL, a short description of the organizational structure will be made.

PN primarily operates in Scandinavia, more specifically Denmark and Sweden. The reasoning for this is the merger of the Swedish postal services (Posten AB) and the Danish postal service (Post Danmark) under the new name PostNord. [PostNord, 2022]
As this project is developed in collaboration with PNL in Denmark, the project will be delimited to the Danish operations.

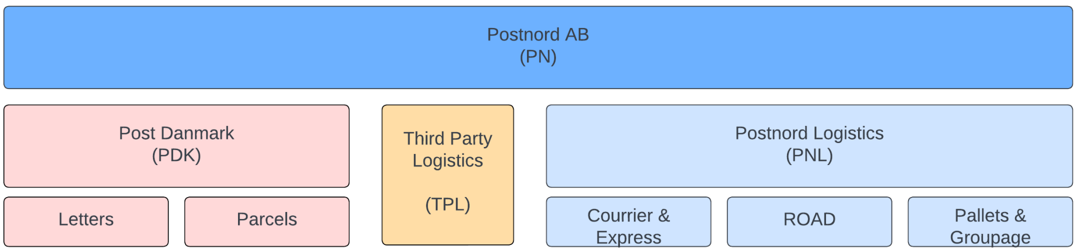


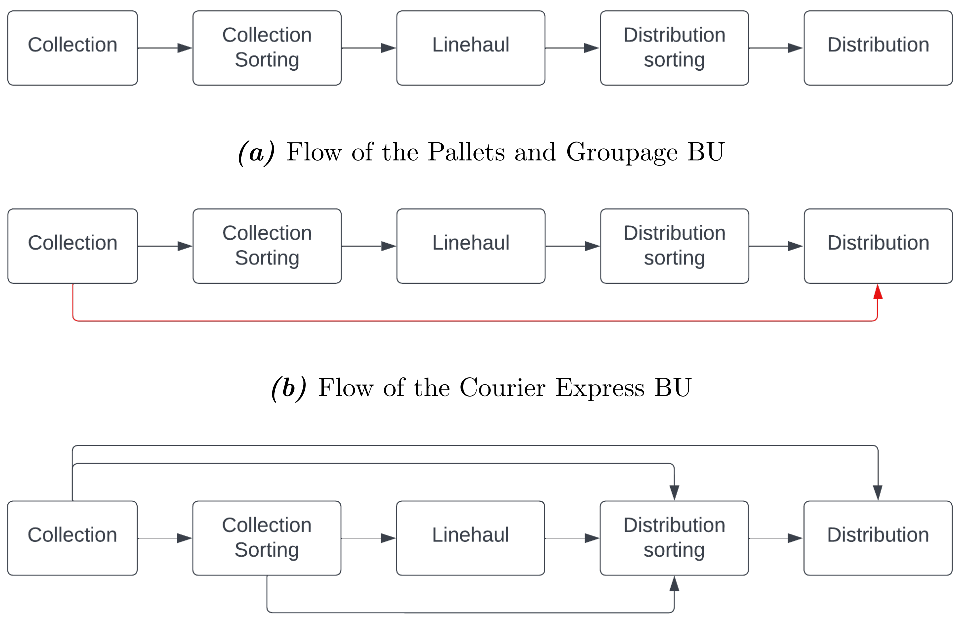
Figure 1.1. Overview of PostNord and subsidiaries in Denmark [Assad Mohammed, 2023]

As seen on Fig. [1.1], there are three subsidiaries, that serve different functions. Post Danmark is responsible for the distribution of letters and parcels under 35 kg to consumers. [Assad Mohammed, 2023] The Third Party Logistics subsidy offers an alternative product where companies can outsource their logistic activities. This includes storage, distribution, import, shipping, and customer handling.[Postnord, 2023]

PNL is generally responsible for B2B deliveries and parcels over 35 kg to consumers. PNL operates three branches being; Courier and Express (CE), Road (R), and Pallets & Groupage (P&G). Pallet and Groupage are responsible for shipments of up to 14 pallets, Road handles shipments of more than 14 pallets. Courier and Express handle priority orders (Express) and In-Night deliveries which are time-constrained and must be delivered before 7 AM. [Assad Mohammed, 2023] As stated earlier, this project is developed in collaboration with PNL, therefore this project will be delimited to focus on the PNL subsidy.

1.2 Orderflow

This section introduces the different order flows that PNL handles. These can vary depending on the quantity and type of order.



(a) Flow of the Pallets and Groupage BU

(b) Flow of the Courier Express BU

(c) Flow of the Road BU

Figure 1.2. Order flow of the different Business Units within PNL [Assad Mohammed, 2023]

Fig. [1.2] illustrates the different order flows that the different departments of PNL handle. A common flow for all departments can be established as; collection of goods, sorting, Line Haul, distribution sorting, and distribution. However, the figure also illustrates diversions from the ordinary flow for both Currier Express and Road. Fig. [1.2b] illustrates the Currier and Express flow. The ordinary flow is used to deliver the Innight parcels that must be delivered before 7:00 whereas traditional deliveries are expected to be delivered between 8:00-16:00. The red line indicates PNL's express deliveries where PNL function as a Currier and thereby picks up a load and directly delivers it to the consumer. Hence they require no sorting or use of the DC capacity. Lastly, Fig. [1.2c] have a variety of order flows due to their operational function as a support unit for the BU's.

1.2.1 Cost distribution

This section seeks to illustrate the cost distributions of the different operations within PNL, furthermore, it seeks to define the problem area for the project.

Initially, semi-structured interviews were conducted to establish a general understanding of the operations and areas of interest. The logistics industry, both for cargo and parcels, has very low-profit margins and is highly competitive. This means that cutting costs is a top priority, even more so than other industries. [Assad Mohammed, 2023]

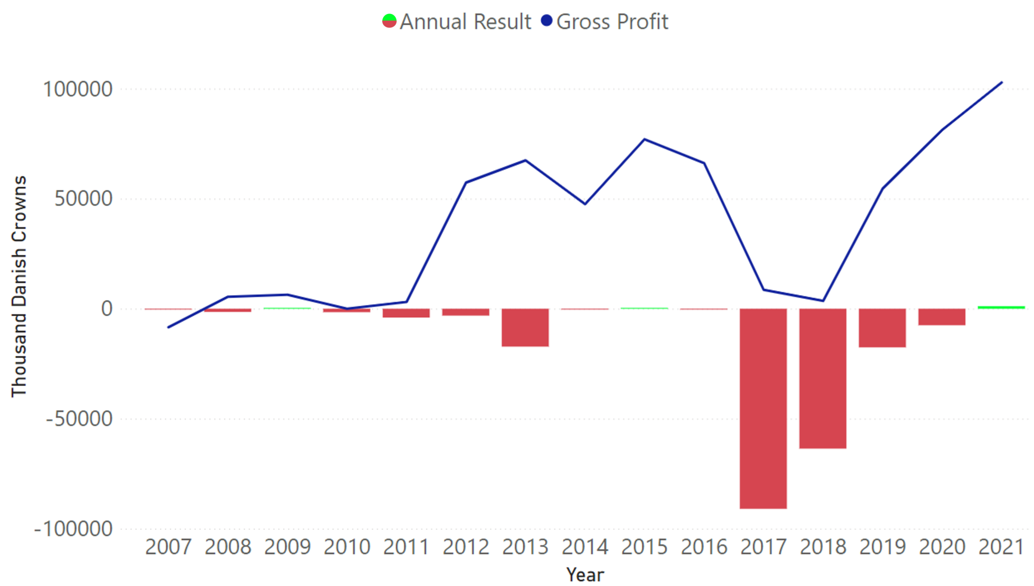


Figure 1.3. Gross profit and Annual Results of PNL (2007-2021) [VIRK, 2022]

Fig. [1.3] illustrates the operational performance of PNL. The figure has been created based on the financial reports of the firm. It can be seen that PNL has operated at an Annual deficit in recent years, however, 2021 managed to present a positive annual result. From 2007 till 2021 PNL has presented 3 positive annual results, which indicates that the company struggles to compete financially. However, it can be seen that the gross profit has increased significantly since 2018 which indicates that PNL initiate initiatives to improve the financial situation. As the revenue is not published it is not possible to convert the gross profit to a profit margin, and thereby comment upon PNL's ability to generate raw profit from their operations. With that being said, Gross Profit is still a great measure, that indicates what a company is left with, after accounting for direct costs of production.[Novy-Marx, 2013]

In conjunction with low profit-margins in the industry PNL operates in, the gross profits and yearly results throughout the years have been slim, as seen in Fig. [1.3]. Considering these financial incentives cost optimization is of great importance to PNL.

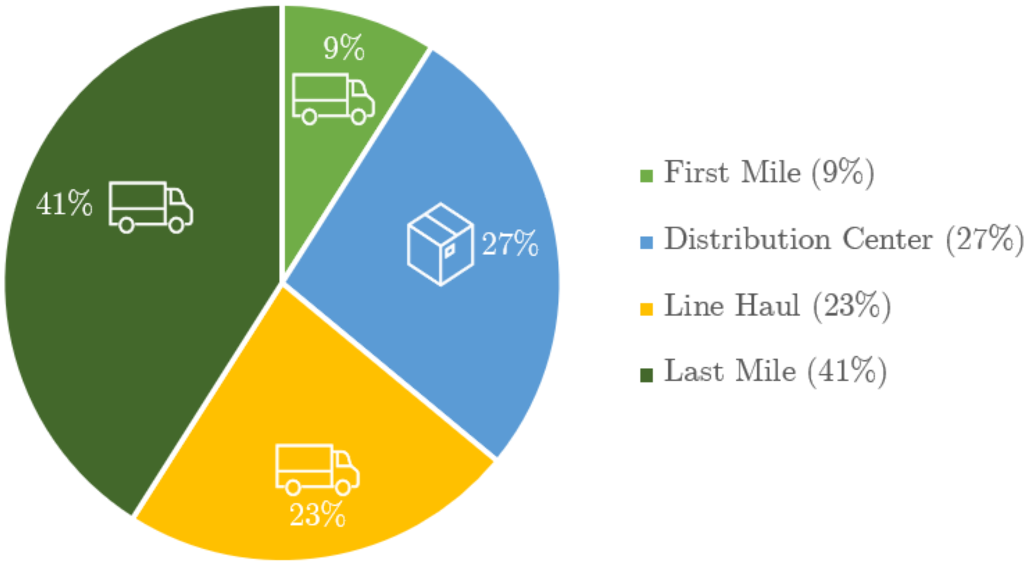


Figure 1.4. Cost Distribution of PostNord Logistics [Assad Mohammed, 2023]

Fig. [1.4] illustrates the cost distribution of operations in PNL. Hence it does not include maintenance, rent, and other fixed costs. It can be seen from the figure that there are three different transportation costs: First Mile, Line Haul, and Last Mile. These constitute 9%, 23%, and 41% of the costs respectively. A total of 73% of the costs are related to route planning and transportation, where Last Mile is the greatest cost factor.

However, as the authors of this thesis have previously written a report for PNL on the subject of last-mile route optimization, this thesis will be delimited from the topic of route planning.

In Fig. [1.4], the second highest expense is the DCs, accounting for 27% of the total costs associated with PNL's operations. Through interviews with PNL, it became apparent that the strategic placement of the DCs was well-considered, whereas the operational flow was determined, primarily without consideration of the order flow. [Assad Mohammed, 2023] Hence it was chosen to delimit the project to focus on the DC operations.

1.3 Initial Problem

The purpose of this section is to introduce the challenges that this thesis aims to address. This is based on the knowledge of the operations gained through interviews outlined in this chapter.
The first challenge relates to the internal flow of the DC pertaining to the layout of the DCs. According to Assad Mohammed [2023], it is suspected that DCs do not align with the actual order flow, meaning that the current layout of the DCs could lead to operational bottlenecks caused by demand fluctuations. In lieu, PNL maintains a fixed and outdated layout.

The second challenge, which is an overall issue across the operation, relates to the outlined cost associated with operations. Profit-margins in the highly competitive logistics industry, and especially concerning parcels and cargo, are slim. The annual results and gross profits reflect the economic challenges that PNL face, which further incentivizes cost reductions. Considering these challenges, the following chapter seeks to identify how to reduce costs in PNLs DCs, specifically relating to cost reductions by optimizing internal flows and potential savings related to layout.

Pre-Analysis 2

2.1 Distribution Centers

Modernization of warehouses has facilitated modern supply chains, as goods move fast through the chain. Companies compete on quality and service time, which in the logistics sector often is reflected by on-time delivery and short lead times. The function of the warehouse has changed over the years from a storage facility holding inventory 'just in case', to a distribution center (DC) where the main role is distribution rather than storage. The goal is cost-efficiency and fast, frictionless flow to the end-user. Thus, the primary function of warehouses is no longer to hold static inventory but rather the tendency is shifting towards functioning as sortation points where products are consolidated, reorganized, and sent along the supply chain - the goal is to reduce or eliminate static inventory. [Bartholdi and Hackman, 2019]

This is reflected in big logistics providers such as Amazon and others, with same-day or even 2-hour delivery. This means, that logistics providers operating in an industry with low profit margins, succeed or fail by their ability to utilize the space and labor of their DC. As a result, modern warehouses offer more usable space and fewer impediments to product flow, as these improvements reduce operational costs, facilitating efficiencies both upstream and downstream along the supply chain. In other words, optimizing the operations of a DC is a main priority to satisfy the fast movement of goods and lower costs in a highly competitive industry. [Bartholdi and Hackman, 2019]

Thus, it has been established that to achieve better warehouse performance, optimization of the operations must be realized. The greatest operational cost is labor in terms of handling and travel time. [Bartholdi and Hackman, 2019]
Therefore, this section will explore and provide an overview of PNL's DC's.

2.1.1 PNL Distribution Centers

The warehouses operated by PNL are also, by the previous statement in Section [2.1], defined as DCs, as goods arrive for consolidation and further distribution, and not for long-term storage. To provide a better overview of the logistics operations of PNL, the geographic location of their DC's is depicted in Fig. [2.1] and Table [2.1]

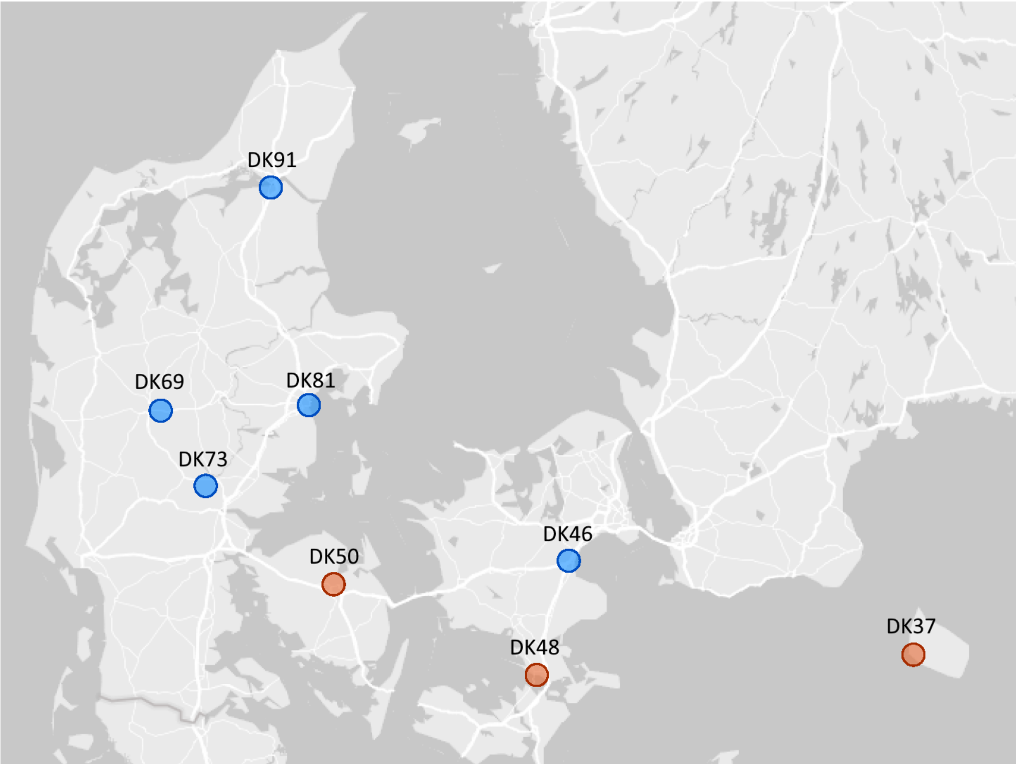


Figure 2.1. Overview of PNL Distribution Centers
[Assad Mohammed, 2023]

As seen in figure Fig. [2.1], PNL operates five DCs in Denmark marked with blue. As described in Section [1.1.1], PNL handles parcels over 35 kg and B2B shipments, primarily larger orders on pallets. Hence the needs of the DCs are very different from the parcel sorting facilities, where machines can sort the parcels on conveyors. The sorting process at PNL is handled manually through the use of electric pallet trucks where the operators empty the trucks, scan the pallets, and move the pallets to a given outbound door based on the destination. The process will be described in further detail in Section [2.3].

Distribution Center	**ZIP**	**City**	**TPL**
DK91	9000	Aalborg	
DK81	8000	Aarhus	
DK73	7000	Taulov	
DK69	7400	Herning	
DK50	5000	Odense	×
DK48	4760	Vordingborg	×
DK46	4600	Køge	
DK37	3700	Rønne	×

Table 2.1. Overview of Geographic Locations of PNL's DCs in Denmark

In addition to the five DCs operated by PNL, as shown in Table [2.1], they have three smaller TPL hubs on Rønne, Bornholm (DK37), Vordingborg, Zealand (DK48), and Odense, Funen (DK50) marked with orange. Through interviews with PNL it became apparent that these DCs have lower throughput and are operated independently. Hence the TPL DCs are disregarded in relation to further analysis throughout the thesis.

2.2 Analysis of Distribution Centers

This section seeks to investigate the DCs of PNL and compare them with the purpose of exploring drivers of cost relating to the DCs and consequently, opportunities to reduce costs.

Data from the DCs have been provided, which will be further examined in Section [2.2.1]

2.2.1 Data Overview

The data used for the different analyses have been obtained through interviews with PNL and subsequent requests. The provided data represents the demand order flow, in other words the throughput of PNL. Depending on the purpose, the data has subsequently been cleaned and filtered. This will be further clarified throughout the sections accordingly. The data stretches from May 2021 to March 2023. In order to provide an overview, in Table [2.2] a sample snippet of the raw data has been provided.

eta_date	**kp2_datetime**	**kp4_datetime**	**product**	**shipment_department**	**delivery_department**	**consignee_place_code**	**kolli**
2022-8-12	2022-8-12, 00:33		Pallet	DK91	DK91	9800	1
2022-1-28	2022-1-27, 16:42	2022-1-28, 04:35	Pallet	DK91	DK46	4200	1
2022-10-12	2022-10-11, 18:19	2022-10-12, 02:52	Pallet	DK46	DK91	9200	1
2022-3-30		2022-3-30, 04:28	Pallet	DK46	DK91	9800	1
2022-8-17	2022-8-16, 16:42	2022-8-17, 02:20	Groupage	DK73	DK91	9240	1

Table 2.2. Sample of Raw Throughput Data from PNL (Index 14-18)

To better interpret the data and its structure, a simple explanation of the columns has been provided.

Eta date

ETA is short for Estimated Transportation Arrival, which means that it is the expected point of retrieval. Once this date has been established PN has an approximate time of arrival, which means that they can start planning the successive processes throughout their delivery system.

Kp2

KP2 is an abbreviation for one of the scans that PN conducts as the cargo moves from place to place. The Kp2 scan is captured as the cargo arrives at a DC.

Kp4

A Kp4 scan is not different from a Kp2, except for the fact that it is captured as the cargo leaves a DC.

Product

Within the product column, there are two different products, namely PostNord Groupage and PostNord Pallet. For PN it is important to distinguish between them, as they represent two different types of cargo. PostNord Pallet is a product code for everything that fits on a pallet, whether it is a quarter, half, or a normal pallet. On the contrary, a PostNord Groupage is anything that does not fit on a regular-sized pallet. Since PN operates within different sectors, PN transports everything from trampolines to large corn silos and this is exactly why it is important for them to separate these product codes.

Shipment Department

The shipment department is denoted as *DK*** and . DK is the country code, and the following numbers represent a DC. Just to give some examples $DK91 = Aalborg$ and $DK81 = Aarhus$ the rest can be seen in Table [2.1]. The shipment Department indicates the first DC the cargo arrives at.

Delivery Department

The delivery department is also denoted as *DK***. DK is the country code, and the following numbers represent a DC. The example from Shipment Department is also applicable to the Delivery Department. The Delivery Department indicates the DC from which the cargo is shipped, onto its final destination.

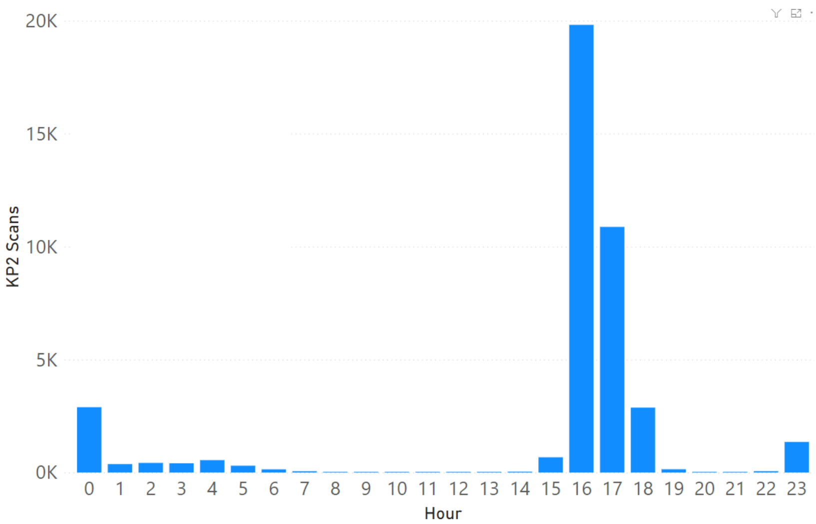
Consignee place code

The Consignee codes are the different postal codes of the consumer.

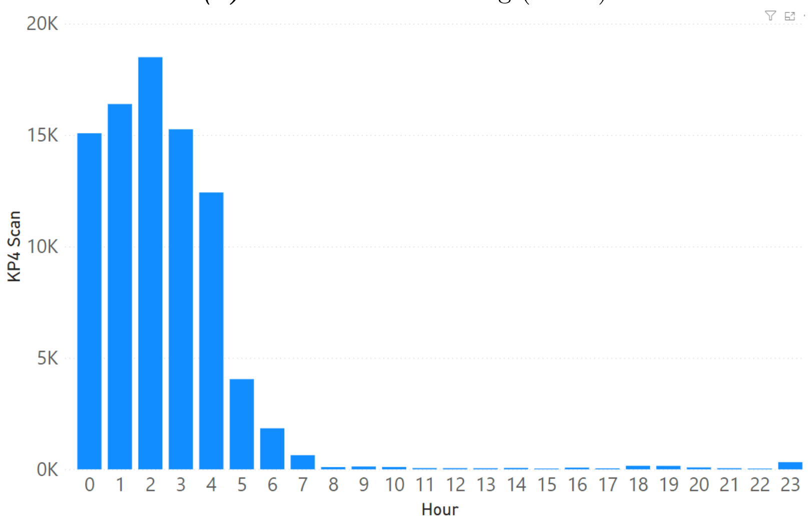
Kolli

Kolli's is the number of shipments that goes onto a single pallet. It is not unlikely that companies have multiple customers that order at the same time. This also means that multiple shipments can be transferred within the same pallet if space allows it. Hence, pallets can contain multiple shipments, and that is why *kolli* can be greater than 1.

As shown in the data sample in Table [2.2], it becomes apparent that some of the KP2 and KP4 scans are missing. Furthermore, the products are supposed to be scanned at DC inception (KP2) and when the product leaves the DC (KP4). Evidently, this is not the case as sometimes, to avoid scanning things multiple times (ex. line-haul Kollis that are to be consolidated elsewhere - they never leave the truck or enter the DC), the product is scanned at another DC. Lastly, it also comes down to human error, as workers quite simply forget to scan products. [Assad Mohammed, 2023] This is a data quality issue that adversely affects further analysis, not only for this thesis but also for the organization as a whole. Specifically, out of all the data (280,542 rows), 87,314 rows or 31.12% of KP2 scans are missing and 18,087 rows or 6.45% of KP4 are missing.
This is important, as with many things, the quality of the input directly affects the quality of the output [Cai and Zhu, 2015].



(a) KP2 Scans in Aalborg (DK91)



(b) KP4 Scans in Aalborg (DK91)

Figure 2.2. Overview of KP Scans on Total Cargo volume

Hence, to reflect on this Fig. [2.2] illustrates the scans. As can be seen, the graphs do not align completely with the working hours specified in Section [2.3.2.1] and Section [2.3.2.2]. However, when PNL was enquired about this, it was explained that scans are not always performed when they arrive at the DC. As mentioned in Section [2.2.1], the scan rate was approximately 69% for the KP2 scan and 93% for the KP4. This indicates that the current processes do not facilitate good documentation practice and inhibit good data quality.

Furthermore, it is stated that the greatest cost factor is labor cost in Section [2.1], however, data is not available for how many workers are in use per hour or even per day, this limits alterations in work schedules, and analyzing number of workers.

2.2.1.1 KPI Measure of DCs

Through the use of semi-structured interviews, it became apparent that PNL's DC's vary in size, layout, and efficiency. It was previously mentioned in Section [2.1], that the greatest cost factor in DC's was labor, where handling and travel time are the primary contributors. Therefore it was chosen to further investigate efficiency, due to its relation to labor cost.

Scan throughput is a KPI measure that PNL applies to measure the efficiency of the DC's operation. The KPI is a compound measure, calculated as scans per worker, per hour for every week. In relation to the project, it will be used as a performance indicator to further analyze the DCs

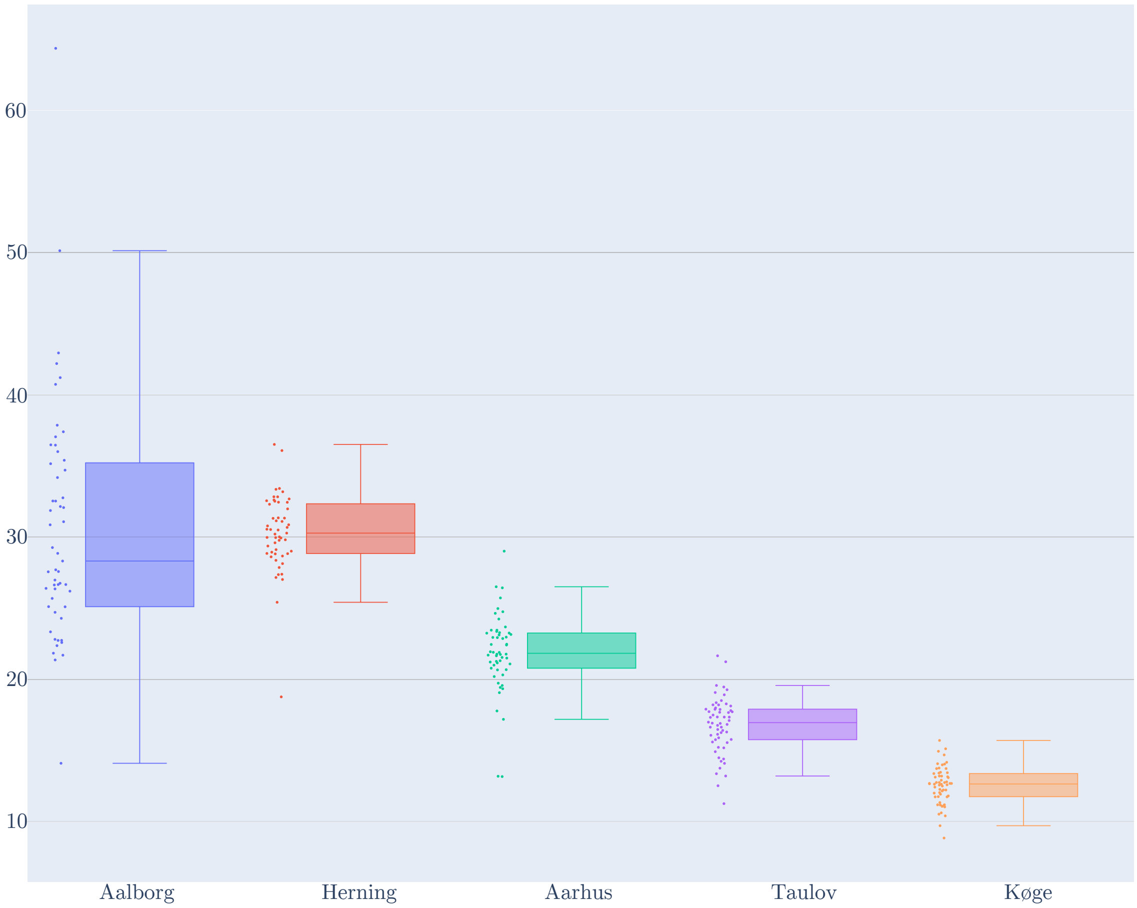


Figure 2.3. Boxplot of Scan Throughput KPI (Week 48, 2021 - Week 1, 2023)

Fig. [2.3] illustrates the efficiency of the different departments of PNL. The box plot illustrates the weekly number of scans, per employee per hour, where each DC has its distinct color. The scans are used as an indicator of efficiency as the products are scanned when they are received and moved to temporary storage. The higher the number of scans, the better the efficiency. It can be seen that Aalborg and Herning DCs are the most efficient. Focusing on the interquartile range, it can be seen that the Aalborg department has more variation compared with Herning and is, therefore, more unstable. It can be argued, that in the observed time period, the Herning DC is the most efficient based on median, even though Aalborg exhibits better efficiencies on individual data points, but has a lower median. This conflicts with PNL statements that Aalborg is more efficient, meaning that over a different time period, Aalborg could be more efficient. However, the given data exhibits fluctuations in efficiency measures of the Aalborg DK91 DC.

Section [2.1] The variation in the scan throughput KPI, which is the efficiency measure, is apparent from DC to DC. Aalborg (DK91) is one of the most productive but also has the highest variance. According to Assad Mohammed [2023], the inconsistency of the KPI span of DK91 is due to demand fluctuations. This will be further investigated in Section [2.2.2].

2.2.2 DK91 KPI Fluctuations

This section seeks to further investigate the KPI span of the DK91 DC. When PNL was inquired about the finding and the apparent gap, especially compared to the four other DCs, it was stated that it was due to demand fluctuations. This is reasonable, as demand is intrinsically part of the measure (a steady demand rate would imply less 'scan downtime'). However, this would imply that the other DCs do not experience demand fluctuations. So to gain further insights into this, the demand of DK91 will be investigated.

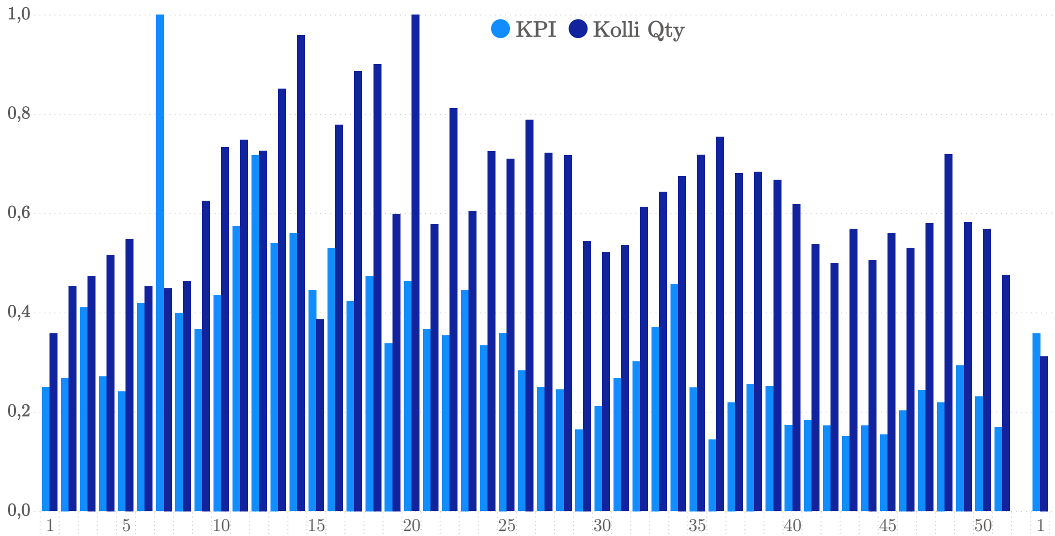


Figure 2.4. DK91 Comparison: Normalized Scan Troughput KPI and Demand (Week 1, 2022 - Week 1, 2023)

In Fig. [2.4], the Scan Throughput KPI and Demand of DK91 is illustrated. In order to compare the data, a linear transformation by min-max normalization has been performed, with values ranging from 0-1; This is also referred to as rescaling. As the name suggests, normalization is performed to rescale the data, as the KPI and demand are measured in different units and consequently in different scales. [Friedman and Komogortsev, 2019; Saranya and Manikandan, 2013]

The bar chart in Fig. [2.4] seems to exhibit some patterns of similarity. To better empirically evaluate how these variables relate to one another, a correlation analysis has been performed in Section [2.2.2.1]. The purpose of conducting a correlation analysis is to test for statistical dependence between two (or more) variables. [Smithson and Popovich, 2003]

2.2.2.1 Correlation Analysis of DK91 KPI

Multiple coefficients of association, or correlation coefficients, exist and have different strengths and weaknesses. In general, the greater the correlation value, the stronger the association. [Smithson and Popovich, 2003]

Correlation Value	**Association**
0 - 0.19	Very Weak
0.2 - 0.39	Weak
0.4 - 0.59	Moderate
0.6 - 0.79	Strong
0.8 - 1	Very Strong

Table 2.3. Level of Correlation [Ratner, 2009]

In Table [2.3], the level of correlation and its interpretation is provided. In the table, only positive values have been given, however, depending on the correlation coefficient, the association can also have a direction; either negative or positive. It must also be stated, that a strong association does not imply a direct connection between the two variables, i.e. correlation does not imply causation. This is often referred to as spurious correlation or correlation fallacy. [Ratner, 2009]

As previously mentioned, there is a wide variety of correlation coefficients to select from and how to select them. It is up to the user and the use case in question. By surveying and understanding the underlying variables of interest. With this in mind, four correlation coefficients have been considered: Pearson's r [SciPy, 2023*c*], Kendall's τ SciPy [2023*b*], Spearman's ρ [SciPy, 2023*a*], and lastly ϕ_K (Phi$_K$) [Baak et al., 2020].

The three formerly mentioned are well-established correlation coefficients, where the latter is a relatively new correlation coefficient with many advantages:

- **Pearson's *r*** measures the linear strength and association between two interval variables. Pearson's r is parametric, meaning it assumes a normal distribution of the data. [Wright, 1921]

- **Kendall's $\boldsymbol{\tau}$** measures the monotonic strength- and direction of association between two ordinal variables and is therefore refereed to as a rank-correlation coefficient. It is non-parametric, meaning it does not assume an underlying distribution of the data [Smithson and Popovich, 2003]. Specifically, this thesis applies Kendall's τ-b, which pertains to how rank ties are handled. [SciPy, 2023*b*]
- **Spearman's $\boldsymbol{\rho}$** is similar to Kendall's, as it is also a rank-correlation coefficient with similar features [Smithson and Popovich, 2003]. It has been disregarded in this thesis, as it is not suitable for sample sizes < 500. [Zwillinger and Kokoska, 2000]
- $\boldsymbol{\phi_K}$ is capable of measuring non-linear relationships between two or more variables of mixed types, and is therefore capable of measuring association between interval, ordinal and categorical data. It cannot measure the direction of association and is more suited for larger sample sizes. [Baak et al., 2020]

Even though the underlying data does not necessarily follow a normal distribution, Pearson's r has still been included. This relates to the interpretability of the measured association, as the distribution affects the p-values and therefore the significance of any measured correlation. [Kowalski, 1972; SciPy, 2023*c*]

The selected correlation coefficients has been calculated in Python by the `scipy.stats` and `phik` libraries, and the source code can be found in Appendix [E.1]. The results and an overview of the correlation coefficient features can be found in Table [2.4].

Coefficient	**Linear**	**Type**	**Direction**	**Distribution**	**Correlation**	**p-Value**
Pearson's r	×	Interval	×	Normal	0.3343	0.014
Kendall's τ		Ordinal	×	Non-parametric	0.2513	0.008
ϕ_K		-		Non-parametric	0.95	2.8

Table 2.4. KPI and Demand Association: Correlation Coefficients and Significance Results

For Pearson's r and Kendall's τ, the significance level $\alpha = 0.05$ has been selected, which means if the p-value $> \alpha$ the null hypothesis (H_0) cannot be rejected, and the measured association is significant. In other words, the null hypothesis states that there is no relationship between the two variables of interest. [Smithson and Popovich, 2003]
For ϕ_K, a modified χ^2 is used instead, where the measured association is significant with values > 5, as the significance level saturates around 5. [Baak et al., 2020]

With this information, it is possible to distinguish a weak association between the scan throughput KPI and the demand of DK91. As seen in Table [2.4], a high correlation value is observed with ϕ_K, but the p-value determines the association is not significant. Pearson's r and Kendall's τ results provide lower correlation values, but the p-values are significant. Considering that Pearson's r is parametric, the significance value might not be accurate. This leaves a significant, weak association between the scan throughput and demand.

A weak association can be interpreted as: when demand goes up, KPI goes up, but this relationship is weak; as the frequency and amount, in terms of span, of this occurrence is not the same from point to point. Even though there is a relationship, this does imply causation, as stated earlier, meaning the demand is not necessarily the cause of a change in the KPI measure. This does not align with PNL's statement that demand is linked to the KPI measure [Assad Mohammed, 2023].

The reason we observe low or non-significant association could be due to the ecological inference introduced by the scan throughput; as the measure is calculated as scan per day per worker per working hour per week, making it a compound measure that can be difficult to decipher and analyze. Ecological inference occurs when comparing an aggregated variable to an 'individual' variable, causing distortion and inaccuracies in the results. [Smithson and Popovich, 2003; Schuessler, 1999]

Thus, the demand and the efficiency KPI fluctuations cannot be empirically linked with the provided methods, but it is suspected that idle workers, and therefore number of workers, in times of low- to no demand could be the reason for KPI deviations for DK91, therefore making it a scheduling issue, as demand on a given day is not always known. [Assad Mohammed, 2023]
This, however, does not explain the KPI variance between the different DC's which mostly exhibits a comparable variance of KPI, when comparing the boxplot in Fig. [2.3]. This is further investigated in Section [2.2.3].

2.2.3 DC KPI in Relation to Size

According to a study conducted by Hackman et al. [2001], it was found that size directly affected efficiency. The study was conducted on 57 DC's and warehouses across various industries. Here it was found, that smaller warehouses tend to be more efficient. The idea is, that the greater the size of a DC facility, as it has to accommodate a greater demand, the greater the handling and travel time, as products have to be moved over greater distances. Furthermore, a DC that can accommodate more demand and products, requires more accessible floor space further increasing the facility size. [Hackman et al., 2001; Bartholdi and Hackman, 2019]

To test the findings of this study with PNL's DCs, the boxplot depicting KPIs has been overlayed with facility sizes for each DC in Fig. [2.5]

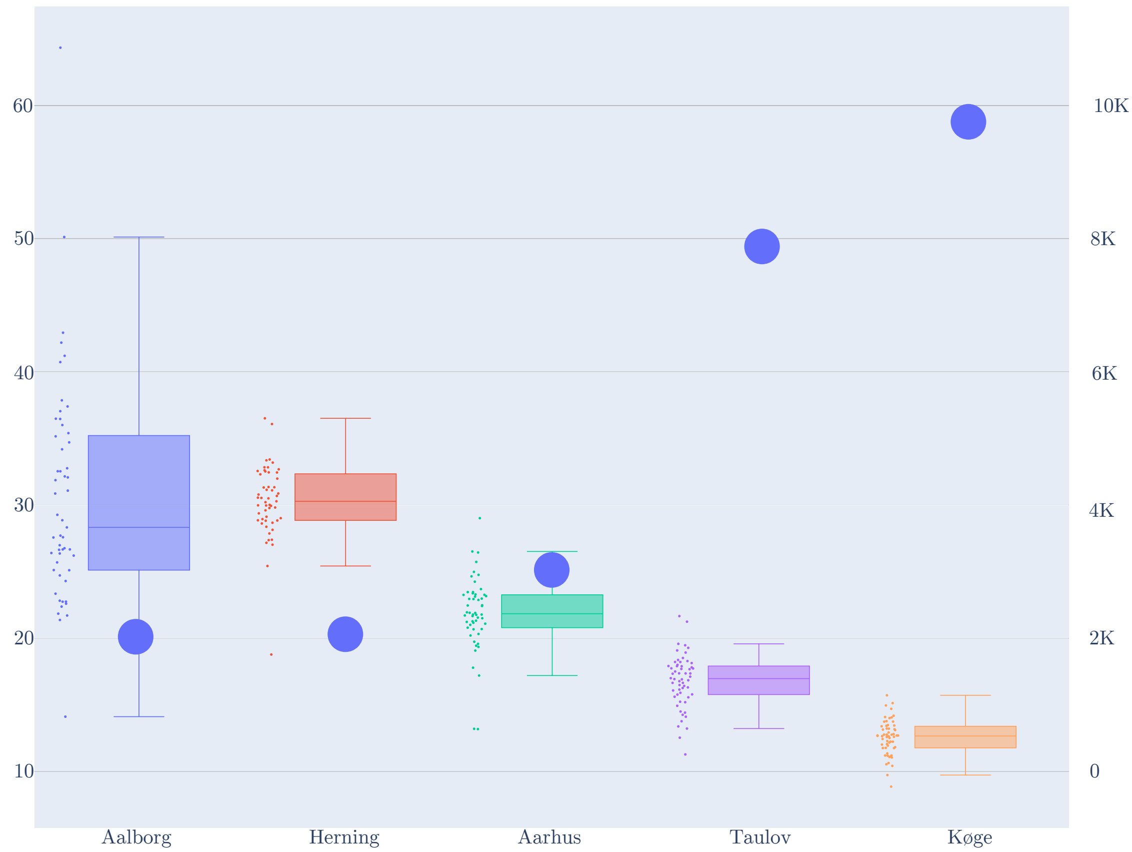


Figure 2.5. Boxplot of Scan Troughput KPI (Left Y-Axis) With Size m^2 Overlay (Right Y-Axis)

DC	Aalborg (DK91)	Herning (DK69)	Aarhus (DK81)	Taulov (DK73)	Køge (DK46)
Size m^2	1952	1977	3048	7905	9806

Table 2.5. Overview of DC Sizes

By observation from Fig. [2.5] and Table [2.5], it is clear to see a relationship between facility size and the efficiency KPI, where the smaller the DC the greater the measured efficiency. This corresponds to the findings of the previously mentioned study conducted by Hackman et al. [2001].

Through interviews with PNL these results were discussed and possible reasons for the variation were identified. It was confirmed by PNL that the processes performed at the different warehouses are similar [Assad Mohammed, 2023]. This leads to the finding, that in order to increase DC efficiency, labor-related costs (both monetary and time-related) should be reduced.

2.2.4 Selection of Distribution Center for Further Analysis

The selection has been based on the findings throughout this section and interviews with PNL, as the associated company contact could provide insights into the key differences between the departments. Through the interviews, it became apparent that the Aarhus (DK81) had a unique L-shaped layout which meant that the operational flow, DC placement, and processing differed from the remaining departments. Hence it was chosen to delimit the DK81 department.

The remaining departments all have a similar layout, a rectangular layout with ports surrounding the facility. The key difference between these departments is the size of the facilities, which results in longer transportation times for the products in the DCs.
As the interviews did not indicate any overall changes in the processes, it was chosen to initially focus on the Aalborg department for multiple reasons.

- The company contact person is situated in Aalborg and can provide easy access to the facility when needed and relevant staff members when needed.
- Aalborg was seen as ideal due to its smaller size in comparison to e.g. Taulov which makes it ideal to analyze and map the flow of cargo.
- Furthermore, it was found in Fig. [2.3] that the Aalborg DK91 DC had the largest variance of the KPI efficiency measure.

Thus, the operations of the DK91 DC and cargo flow will be further examined in Section [2.3]

2.3 Operations

This section includes an analysis of the cargo flow and decision processes for the Aalborg Distribution center. Furthermore, it includes an analysis of the throughput, describing the inbound and outbound flow.

To illustrate the current process flow in PNL it was chosen to create a flowchart. A flowchart can be a useful tool to illustrate a process visually and can be useful to identify where problems occur. The meaning of each shape in the flowchart can be found in Fig. [2.6], and the flowchart in Fig. [2.7].

Figure 2.6. Overview of Flowchart Shapes and Meaning

The flow chart was created through interviews and observations and validated in consensus with PNL.

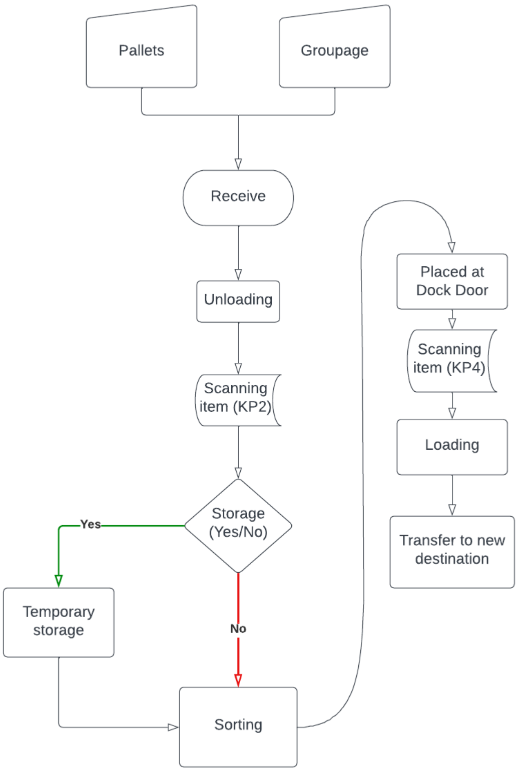


Figure 2.7. Flowchart Illustrating Cargo Flow at a DC

2.3.1 Distribution Analysis

This section presents data for the outbound flow of the DC. The department in Aalborg is responsible for the last-mile distribution in most of Northern Jutland, however, they also serve as a DC to distribute goods from Northern Jutland to the rest of the country. The figure further illustrates the demand for the rest of the country. However, instead of illustrating the consignee codes, the demand has been consolidated at the local DCs.

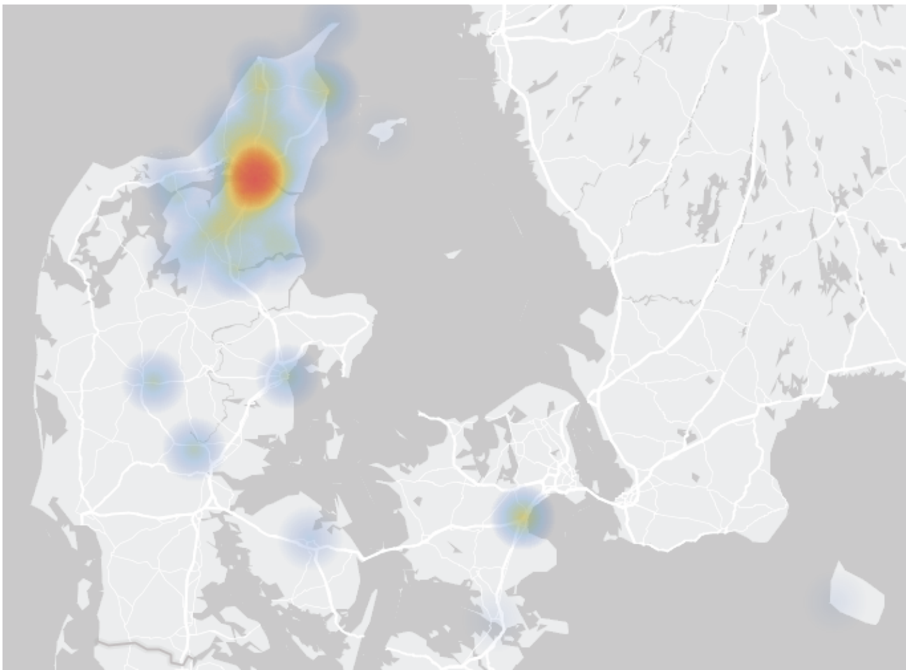


Figure 2.8. DK91 DC Heatmap of demand from February 2022 - February 2023

The Fig. [2.8] illustrates the demand locally in Northern Jutland, as every consignee is represented by ZIP code. The delivery patterns have been selected in the time period from February 2022 to February 2023 to exclude potential demand pattern discrepancies from COVID-19. For the remaining parts of Denmark, the shipments are consolidated at their respective delivery DCs.

Fig. [2.8] illustrates demand locally in the DK91 area whereas the remaining orders are placed at the last DC prior to delivery. Hence, orders that are to be delivered in Zealand are transported to the Køge DC (DK46).

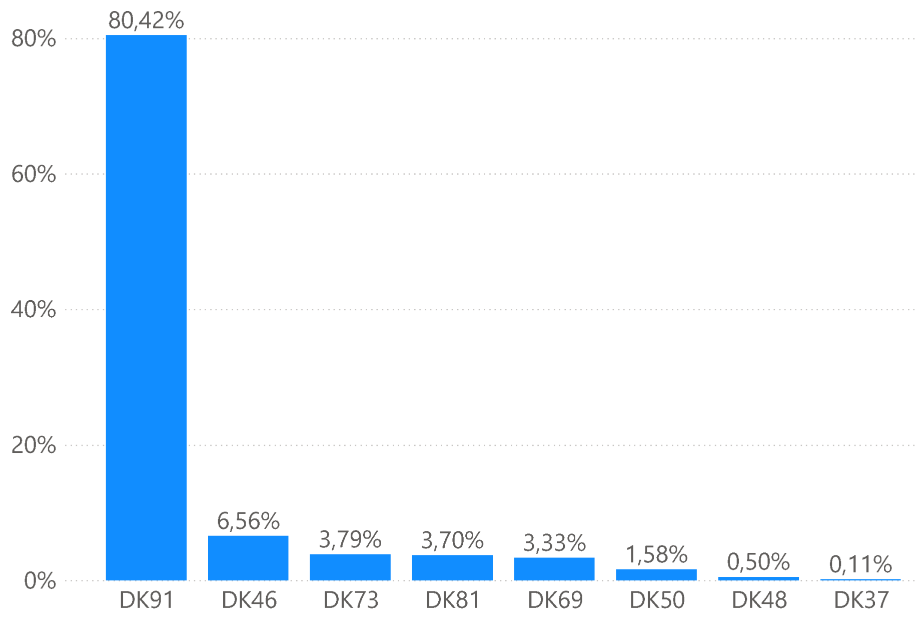


Figure 2.9. Demand Distribution for DK91 (Aalborg)

Fig. [2.9] illustrates the demand for the Aalborg department. It can be seen that 80% of the demand is distributed in the local area. The second largest recipient of goods from Aalborg is the Køge (DK46) department.

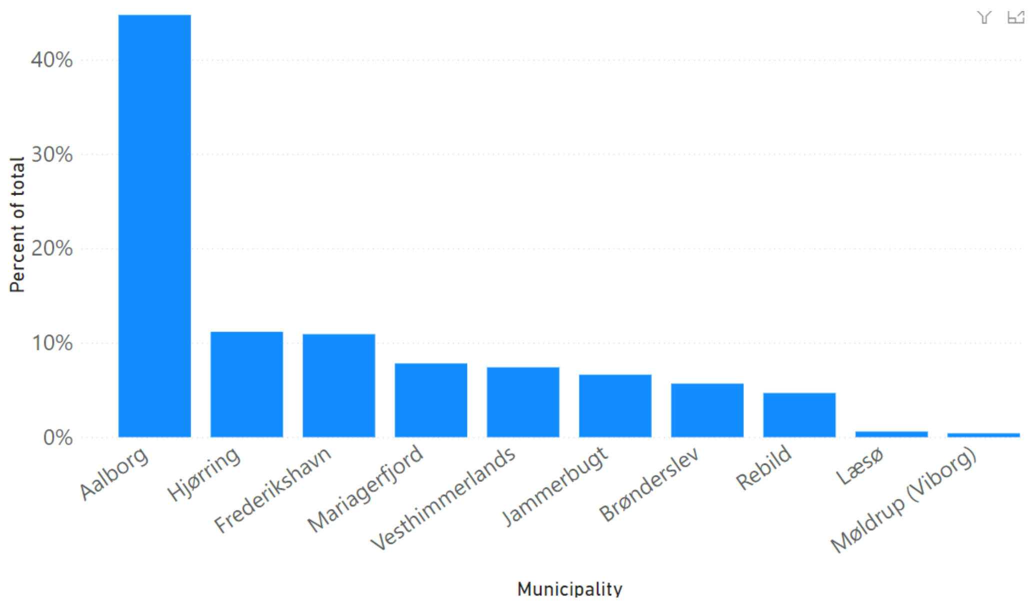


Figure 2.10. Distribution of Demand Sorted by Municipalities

To quantify these tendencies Fig. [2.10] illustrates a percentage value of the total colli delivered from the Aalborg Distribution center. Most notably, 45% of the delivered collis, are delivered within Aalborg Municipality. Furthermore, it illustrates how much of the DC capacity is used for the different municipalities with regard to both handling and storage.

2.3.2 Distribution Center Flow

This subsection introduces the operational flow within the PNL DC in Aalborg. This includes a description of the two different arrival patterns and their differences in handling.

It is important to understand the worker's role in the facility. Ultimately their responsibility is to sort the goods in a timely manner to allow the trucks to transfer and distribute the goods. There are two types of inbound goods, being locally collected and Linehaul. Linehauls are trucks that transfer goods between DCs, hence the goods that arrive at the Aalborg DC through linehaul, are shipped from other regions in the country. The locally collected goods, on the other hand, can have a consignee place code in other regions of the country, which means that the product needs to be Linehauled to another DC. As a result, PNL operates two shifts with different responsibilities. These will be described below:

2.3.2.1 Preliminary sorting

From 14:00 to 21:00 the Preliminary Sorting team is responsible for the initial sorting of the locally collected goods. These arrive throughout the day and include goods that are to be distributed to the entire country. Therefore the Preliminary Sorting team is responsible for a "rough" sorting process where pallets are placed in different zones based on their consignee place code. If the product is to be delivered in e.g. Aalborg, it is placed in DK91, whereas Aarhus is placed in DK81, etc. As can be seen on Fig. [2.11], there are four storage areas. These are determined based on the Linehaul routes. *DK81/DK69* is the Midtjylland region and *DK37/DK48/DK50/DK73* is the remaining part of Denmark. Lastly, the *NO* storage area handles demand from Norway and is thus delimited from this thesis. Every zone except for the DK91 is shipped directly from the temporary storage zones. However, the DK91 goods need additional sorting, which is performed by the Secondary sorting team.

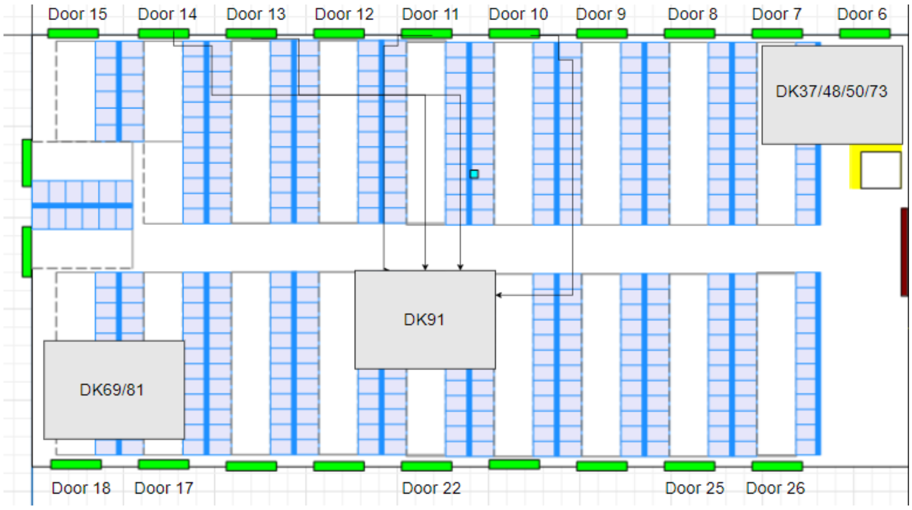


Figure 2.11. Preliminary Sorting at the Aalborg DC

2.3.2.2 Secondary sorting

The Secondary sorting team, works between 23:00-08:00, with the primary responsibility of ensuring that every pallet is placed at the correct dock door before 08:00. The team has two tasks, firstly to distribute the goods at the DK91 Preliminary Sorting zone to the dock doors. The other task is the inbound linehaul from the other DC's in Denmark. This linehaul contains cargo to be delivered in DK91, hence there is no need for the preliminary sorting process. The Secondary sorting team places the DK91 pallets based on the consignee place code. Every dock door has specified postal codes listed on the door which is based on the delivery routes. This means that am item for delivery in e.g. 9000 Aalborg, always needs to be placed at the same dock door.

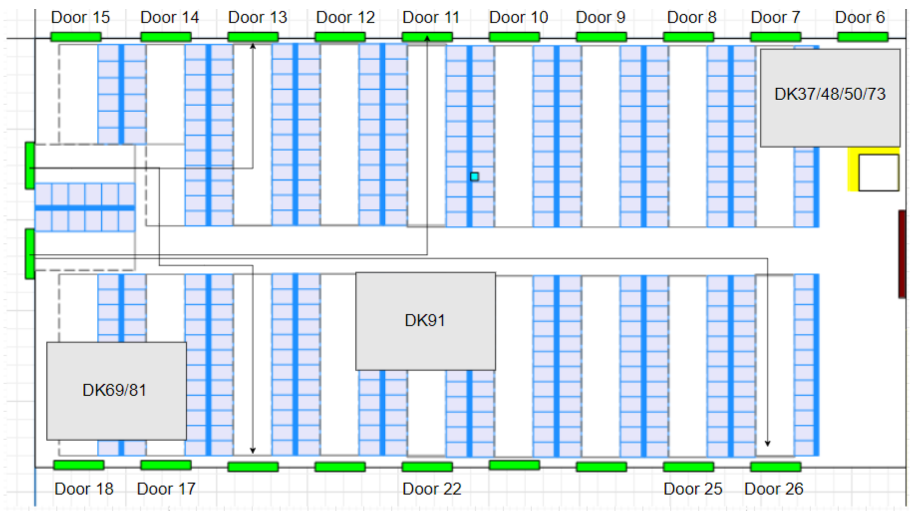


Figure 2.12. Secondary Sorting at the Aalborg DC

2.4 Chapter Conclusion

Throughout this chapter, multiple analyses have been conducted to further investigate the DCs of PNL. The purpose of this was to explore how cost reductions of the internal flow of PNL's DCs could be achieved.

This led to a correlation analysis of the demands' impact on efficiency, however, a relationship could not be empirically linked. Instead, comparing DC size to respective KPI measures, in Section [2.2.3], it was found that there is a relationship between the size of the facility and the PNL efficiency measure. Hence the larger the facility the lower the efficiency. Aalborg (DK91) was selected for further analysis, due to several factors including its variance in efficiency.

In Section [2.3] the operational flow of the DC was analyzed and mapped. The tasks are distributed to two teams, who work different shifts. The first team named Preliminary Sorting prepares cargo that are collected in North Jutland for further sorting in DK91 and for Linehauls to the remaining DCs. The second team named Secondary Sorting is responsible for further sorting of the DK91 temporary storage and the DK91 cargo that arrive through line hauls.

It was found that the configuration of the DC was static and did not take varying demands into account; such that specific postal codes are always placed at the same dock door. This means, that if demand is low for one of the dock doors nearest to the inbound, not only is the dock door under-utilized, but the remaining dock doors that are farther away have to travel longer distances. This could lead to increased handling time and distances traveled, and therefore, less efficiency.

3 Problem Statement

As stated throughout Section [1.2.1], PNL operates in a competitive market with slim profit margins, and annual results revealing a financially struggling business. This renders optimizations to reduce costs a critical factor for retaining competitiveness for PNL. It was found that for DCs, the greatest cost factor is labor cost, meaning increasing efficiency would decrease costs. It was identified that KPIs across the PNL DCs varied in relation to size, which corresponds to the tendency, that handling and traveling time to increase as DC size increases. Even though the Aalborg (DK91) DC was among the best performing, by comparing the KPI efficiency measure. However, DK91 had the greatest variance, meaning it was selected for further investigation. Due to the circumstances of being a logistics provider, PNL and specifically the Aalborg (DK91) DC has demand fluctuations, which in turn requires flexibility in the day-to-day operation. It was found, that the DC operations had two flows, one for preliminary sorting and one for secondary sorting, meaning that some colli are handled twice before being sorted to their dock door. The dock doors have a rigid allocation, meaning that each demand group is always allocated to the same location. Taking these findings into consideration, the problem statement is formulated as:

By analyzing the DK91 DC in Aalborg, how can the efficiency and flexibility of PNL DCs be improved?

- How does the internal flow affect efficiency and flexibility in PNL DC operations?
- How can the DC operations be improved with regard to efficiency and flexibility?
- How can an alternative to the current state be implemented?

Analysis: Part I 4

This chapter seeks to further investigate the findings of Chapter [2], and answer the problem statement. This will be done through further analysis of the DC operations and a solution proposal. The goal is to identify methods to reduce travel and handling time in relation to increasing DC efficiency.

4.1 Current Operations

This section seeks to analyze the current operations further, as to identify indicators that inhibit efficiency. This will be done through measurements, interviews, and use of data.

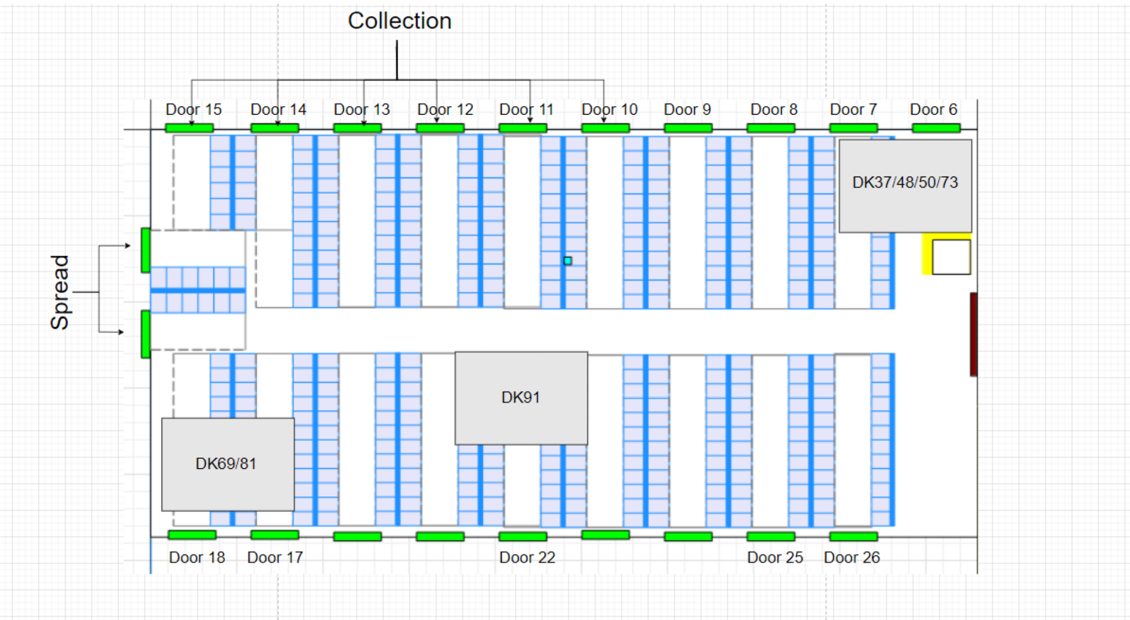


Figure 4.1. Dock Doors Used for Cargo Distributed in Northern Jutland

As seen in Fig. [2.7], PN DC's have several flows, which also increases the complexity when they are further analyzed. Different groups of cargo are received and processed throughout the entire day, depending on both their origin and destination. Due to this fact, the section will include an analysis, where the two different flows described in Section [2.3.2] will be used to further analyze the dock door setup. To better illustrate the cargo sorting and handling process, each dock door has been enumerated and grouped, as seen in Fig. [4.1]. As the abbreviations such as DK91 and DK81 cover a multitude of different postal codes, a table has been generated to create an overview of the different postal codes, as well as the designated dock door

Dock Door	Group	Postal Codes	Dock Door	Group	Postal Codes
15	**1**	9990, 9981, 9982, 9970, 9870, 9881	**7**	**9**	9000
14	**2**	9900, 9940, 9300 9340, 9352	**18**	**10**	9400
13	**3**	9310, 9362, 9370, 9330, 9320	**22**	**11**	9220
12	**4**	9800, 9850, 9700, 9760, 9830, 9740	**25**	**12**	9210
11	**5**	9690, 9460, 9440, 9403, 9480, 9490, 9442, 9443, 9382, 9381	**26**	**13**	9200, 9100
10	**6**	9240, 9670, 9681, 9610, 9600, 9620, 9640, 9632, 9631	**19**	**14**	DK81/69
9	**7**	9270, 9280, 9550, 9560, 9500, 9510,	**6**	**15**	DK37/48/50/73
8	**8**	9520, 9530, 9230, 9260, 9293, 9574, 9575			

Table 4.1. Dock Doors and Their Respective Postal Code Divisions

In Section [4.1], the postal codes for dock doors 6 and 19 respectively have not been split into dedicated postal codes. Since the DC is located in Northern Jutland, most of the dock doors are assigned to serve the different regional parts, and hence the postal codes are important to include for the general overview. The given postal codes for dock doors 6 and 19, can be seen in Appendix [B]. As seen in the table, between 1 to 9 postal code areas are assigned to a single dock door, apart from dock doors 6 and 19 as already described. Here it can also be seen how postal codes are often put into groups for cargo to be consolidated and thereby distributed from the car that operates within that specific area. As soon as the cargo has been received, it is transported from the receiving dock door onto the temporary storage area, and from here, the process gets more complex. As seen in Section [4.1], 15 dock doors account for the total flow of goods that are distributed in both Northern Jutland and to the other DCs. This means, that depending on the cargo's respective delivery address, it is assigned to one of the 15 ports.

In order to further analyze the dock door allocation, the data from Section [2.2.1] has been applied. For this specific purpose, the demand has been summarized according to the demand groups in Section [4.1], resulting in Table [4.3] and Section [4.1]. The total demand is sorted from highest to lowest, with one year worth of demand data.

As described in Section [2.3.2], there are two different flows, namely preliminary and secondary sorting. Essentially, cargo is treated in two different ways, depending on whether it is directly re-distributed or if it is temporarily stored at the DC. Therefore, the demand flow can be split into two sorting processes: Preliminary Sorting (PS) and Secondary Sorting (SS). The preliminary flow handles inbound pallets gathered from Northern Jutland and sorts them based on region. These goods are handled through the first six dock doors from the left as illustrated on the northern part of the facility in Fig. [4.1]. On the left side of the facility, goods arrive from the other PNL DCs through a process named Linehaul. This distinguishes it from PS, as these goods only undergo the SS process and hence demand must be separated across two different sources in the simulation model.

Through data analysis in Fig. [2.2] it was found that KP2 scans represent the arrival of goods in the facility.

Shipment Department	Delivery Department	Process	Local Delivery
DK91	DK91	Preliminary Sorting + Secondary sorting	Yes
$\neq$ DK91	DK91	Secondary Sorting	Yes
DK91	$\neq$ DK91	Preliminary Sorting	No

Table 4.2. Data Subsets: Preliminary Flow and Secondary Flow

Table [4.2] illustrates the decision parameters that were used on the KP2 data. With the purpose of disaggregating the demand into different types. The shipment department indicates the first DC to receive an order, whereas the delivery department indicates the last DC to handle the order prior to delivery. This distinction is important as the process varies depending on said factors. All demand of Shipment Department DK91 needs to go through the preliminary sorting process as they could be bound for delivery in other DCs than DK91. However, if the Delivery Department does not equal DK91, there is no need for SS as this process is only performed for local deliveries. The last process is the goods where the shipment department does not equal DK91, which are delivered from other DCs. These goods only need to undergo secondary sorting.

Dock Door	Total Demand	%-Spread	Dock Door	Total Demand	%-Spread
12	18988	12.2	**22**	11943	7.7
10	17289	11.1	**18**	10834	7.0
7	16894	10.9	**13**	8686	5.6
11	15981	10.3	**15**	6010	3.9
9	15315	9.8	**14**	4684	3.0
26	13247	8.5	**25**	2661	1.7
8	13143	8.4			

Table 4.3. Total Demand for Dock Doors - Secondary Sorting

Dock Door	Total Demand	%-spread	Dock Door	Total Demand	%-spread
6	12858	44.7	**18**	291	1
19	10621	36.9	**7**	278	0.9
12	819	2.8	**13**	271	0.9
11	775	2.6	**26**	200	0.6
10	737	2.5	**14**	184	0.6
9	663	2.3	**15**	164	0.5
8	488	1.6	**25**	51	0.1
22	335	1.1			

Table 4.4. Total Demand for Dock Doors - Preliminary Sorting

Before commenting on the demand from Table [4.3] and Section [4.1] for the different flows, it is important to mention where the cargo arrives from. When considering the SS, all the cargo arrives from the two dock doors on the west side of the building as depicted in Fig. [4.1]. As for the SS, the cargo arrives at six ports on the northern side of the building, namely dock doors; 15, 14, 13, 12, 11, and 10.

To further investigate opportunities for improvement, it was chosen to compare dock door allocation to demand with travel- and handling time in mind. The general assumption is, that if a dock door with high demand, is further from the off-loading zone, a greater distance is traversed and thereby increasing travel and handling time.
Through interviews with the PNL contact person, it was confirmed that the allocation had not been revised for years [Assad Mohammed, 2023]. Therefore, there might be a potential for improving the flow at the DC by revising dock door allocation.

By comparing the dock door setup with the demand in Fig. [4.2], it can be seen, that the dock doors are allocated unevenly compared to their respective demand. Fig. [4.1] illustrates that demand is received through 2 ports in the west part of the facility and 6 in the north end of the facility. Thus it can be seen on Fig. [4.2], that the dock doors with the greatest demand are placed the furthest from the off-loading zone. Therefore this section will explore the potential for a demand-based dock door allocation to reduce the distances and time of the sorting processes.

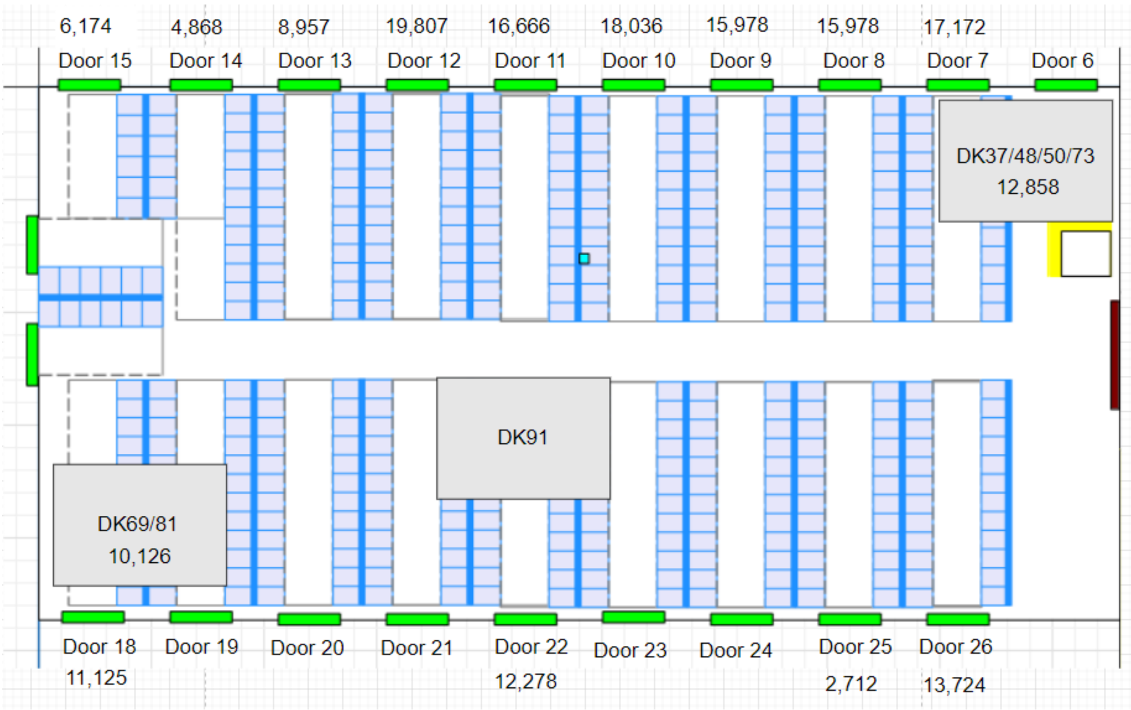


Figure 4.2. Demand per Dock Door (February 2022 - February 2023)

For demand group 14 and 15, these distinguish themselves from a flow perspective, as these groups are only present in the PS flow - as they are picked directly from the sorting zone as depicted in Fig. [4.1]. PS demand for group 14 and 15 is greater than the demand for the other PS groups, accounting for 81.6% of the PS demand. Even though they are a part of the PS process, the cargo for dock doors 6 and 19 is actually stored directly in the gray storage zones. The demand for the remaining dock doors, relating to PS, is temporarily stored in the middle of the DC. The storage zones are depicted in Fig. [4.1].

4.1.1 Demand in current operations

Through interviews with the PNL representative, demand at PNL seems to follow a distribution centered around the mid-week mark throughout the week.

In order to visualize this, the demand for two independent weeks has been plotted in Fig. [4.3].

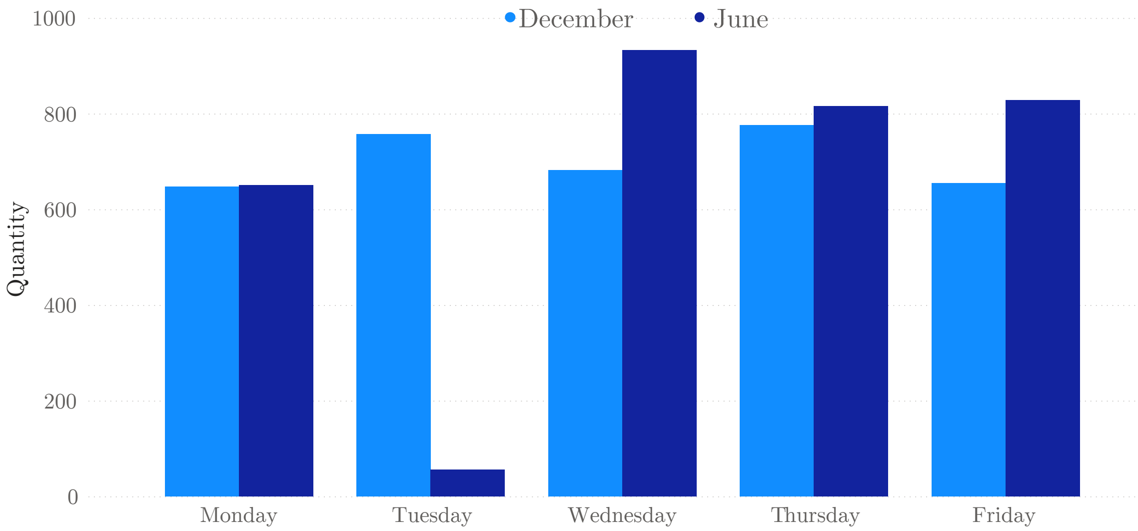


Figure 4.3. Demand for the First Week of June and December 2022

As seen in Fig. [4.3] the demand throughout the weeks are fluctuating. By reviewing Fig. [4.2] and considering the current dock door allocation, it could be assumed that high-demand groups are moved away from the arrival zone in the northwestern corner of the facility. However, through interviews with the PNL representative, it was clarified that the PS and SS demand is spread across two shifts and thus does not interfere with each other; which facilitates opportunities for revising the current dock door allocation.

As shown in Fig. [4.3], the demand is fluctuating throughout the week, and moreover, the logistics market requires even faster processing times with small profit margins, as stated in Section [2.1]. Thus a fixed setup, in terms of dock door allocation, would face the same issues as the current setup. Hence, a fixed setup should not be considered, when the demand changes from day to day. In order to illustrate this, the same weeks of demand have been plotted, while considering the postal code groups from Section [4.1].

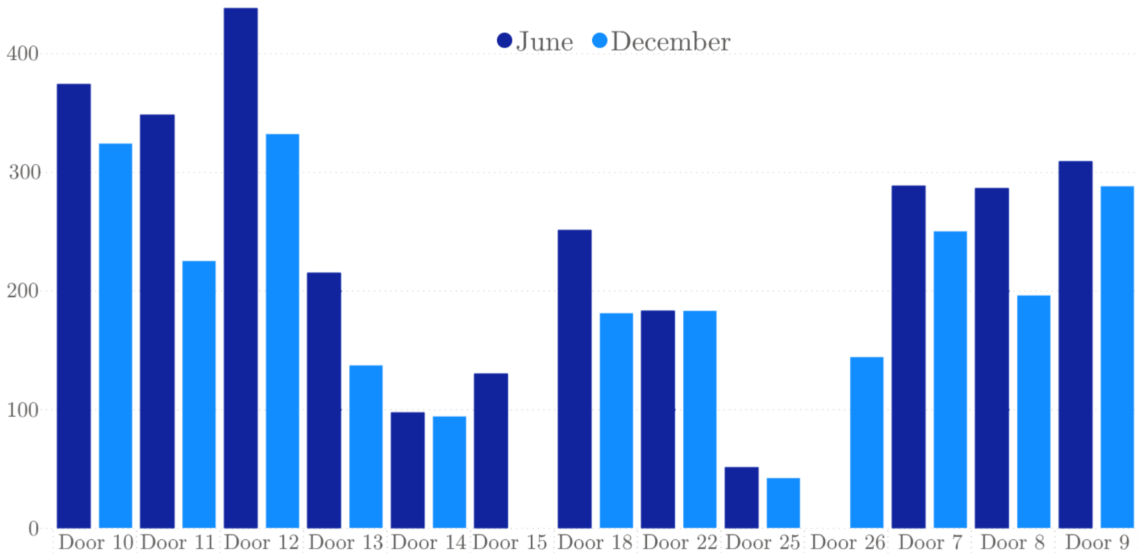


Figure 4.4. Demand Distribution Based on the First Week in June and December 2022 Secondary Sorting

Based on both total yearly demand and weekly demand in regard to different dock doors, a potential to improve performance is identified by allocating existing demand groups to other dock doors. As mentioned in Section [2.1] one of the greatest cost factors when considering DCs is the traveling and handling time. So, in order to further investigate the current setup, the respective distances from the individual ports to the off-loading zone are required.

4.1.2 Distances

This section will seek to quantify and reduce distances by alternative dock door allocation.

The distances have been measured in regard to real obstructions such as markings on the floor or supportive pillars throughout the DC. This was done in order to represent the actual operations within the DC, instead of measuring the distance in a direct line. The distances can be seen in Table [4.5]. As seen, the distances between the off-loading zone and the individual dock doors differ, both in regard to the preliminary and secondary sorting. This further emphasizes the importance of considering which postal code groups to designate to a specific dock door as one variable might dictate one setup, whereas a change in this variable might dictate something completely different. In order to further investigate this, the total distance traveled from the off-loading zones to the dock doors will be presented, for both the preliminary and the secondary sorting. This has resulted in three different tables Tables [4.5] to [4.7]. This is due to the fact that the secondary sorting is only handled once, whereas the preliminary sorting is handled two times; one time from the dock to storage, and one from storage to the dock. This is done for both one year, as well as the individually chosen weeks, that have been presented in Fig. [4.1].

Dock Door	Distance	1 Year	June	December	Total 1 Year	Total June	Total December
12	22,8	18988	438	332	432.926,4	9.986,4	7.569,6
10	32,8	17289	374	324	567.079,2	12.267,2	10.627,2
7	47,2	16894	288	250	797.396,8	13.593,6	11.800
11	27,7	15981	348	225	442.673,7	9.639,6	6.232,5
9	37,2	15315	309	288	56.971,8	11.494,8	10.713,6
26	51,6	13247	0	144	683.545,2	0	7.430,4
8	42,6	13143	286	196	559.891,8	12.183,6	8.349,6
22	32,6	11943	183	183	389.341,8	5.965,8	5.965,8
18	14,1	10834	251	181	152.579,4	3.539,1	2.552,1
13	18,4	8686	215	137	159.822,4	3.956	2.520,8
15	9,2	6010	130	0	55.292	1.196	0
14	14	4684	97	95	65.576	1.358	1.330
25	47,1	2661	51	42	125.334,1	2.402,1	1.978,2

Table 4.5. Distance, Demand and Accumulated Distance Based on Demand for Secondary Sorting

Dock Door	Distance	1 Year	June	December	Total 1 Year	Total June	Total December
15	25	4789	54	67	119.725	1.350	1.675
14	25	4789	54	67	119.725	1.350	1.675
13	25	4789	54	67	119.725	1.350	1.675
12	25	4789	54	67	119.725	1.350	1.675
11	25	4789	54	67	119.725	1.350	1.675
10	25	4789	54	67	119.725	1.350	1.675

Table 4.6. Distance, Demand and Accumulated Distance Based on Demand for preliminary sorting to off-loading

Table [4.6], illustrates how both the distances and the demands are distributed across the inbound dock doors equally. This has been decided due to the lack of data from PNL, which meant that it was impossible to tell which docks the incoming cargo had actually been received at, hence, it was decided to spread the cargo evenly.

Dock Door	Distance	1 Year	June	December	Total 1 Year	Total June	Total December
6	8,4	12.858	158	172	108.007,2	1.327,2	1.444,8
19	4,65	10.621	139	143	49.387,7	686,35	664,9
12	16,1	819	0	0	13.185,9	0	0
11	12,7	775	0	0	9.842,5	0	0
10	17,5	737	0	0	12.897,5	0	0
9	22,5	663	12	33	14.917,5	270	742,5
8	27	488	0	0	13.176	0	0
22	5,3	335	6	11	1.775,5	31,8	58,3
18	25,6	291	4	16	7.449,6	102,4	409,6
7	31,8	278	3	10	8.840,4	95,4	318
13	22,5	271	0	0	6097,5	0	0
26	26,1	200	0	0	5.220	0	0
14	27	184	0	14	4.968	0	378
15	31,8	164	0	0	5.215,2	0	0
25	21,5	51	2	2	1.096,5	43	43

Table 4.7. Distance, Demand and Accumulated Distance Based on Demand for Preliminary Sorting From Off-Loading

When summing up all the distances traveled depending on the dock door setup, the total distances are the following. It is important to note that the numbers from Tables [4.5] to [4.7] has been multiplied by two, as the transporter has to return to the pick-up point as it has processed a shipment. Thus the results are:

- **1 year**: 11,963,564 meters
- **Week in June**: 196,396 meters
- **Week in December**: 182,356 meters

These results are based on sorting the dock door setup, according to historic demand data. Therefore, the dock door allocation depends on the time period, which alters distances are given.

4.1.3 Suggested Operations

The idea is to match the greatest demand with the closest dock door that sits closest to the off-loading zone or arrival dock, depending on the flow. As mentioned, dock door assignment for the respective trucks is not considered, as the cargo information is insufficient. In Appendix [A] it is seen that when the respective demand is filtered, the dock door and thereby postal code groups that produce the most demand change from period to period. Even though some of the high-runners are more likely to occur in the top 3, there seems to be a greater variation in the remaining groups. Additionally, the demand will intrinsically be skewed in the preliminary sorting, since demand group 14 and demand group 15 constitute 81.6% of the PS demand. Therefore Due to this, their position will not be altered, as they are already as close to their respective dock doors as possible Fig. [4.1]. The total distance traveled when considering the proposed setup, where both demand and distance are taken into consideration are as follows.

- **1 year**: 9,655,748 meters
- **Week in June**: 157,302 meters
- **Week in December**: 138,938 meters

From these results, an opportunity for improvement of efficiency can be realized with the proposed dock door allocation; this is achieved by reducing distance, which is reduces both travel- and handling time. The improvements will be calculated in percentage.

$$\text{Percentage decrease} = \frac{\text{Increase}}{\text{Original number}} \cdot 100 \qquad [4.1]$$

Period	Decrease in meters
1 year	19.3%
1 week in June	19.9%
1 week in December	23.8%

Table 4.8. Percentual Decrease in meters

As illustrated in Table [4.8], it can be difficult to quantify the actual impact on PNL in terms of how their daily operations are running. Handling of cargo in the DC, PNL uses electric forklifts. The speed of an electric forklift has been measured in order to convert the decrease in meters into an actual time measurement. The top speed was measured at 10km/h which, can also be confirmed by the manufacturer's own website [Linde, 2022]. But, this is considering free roam, meaning that no obstacles are in the way. This assumption can not be made in a DC environment, since other workers, pallets, and floor markings, will ultimately dictate the pace of the forklift. Therefore, the average speed of the forklift is assumed to be 5km/h according to [Gates, 2021] Firstly, the decrease in meters will be converted to kilometers.

$$\text{Kilometers} = \frac{\text{meters}}{1000} \tag{4.2}$$

So, the initial savings converted to kilometers are the following:

- **1 year**: 1366.31 kilometers
- **Week in June**: 15.08 kilometers
- **Week in December**: 11.24 kilometers

Assuming that the average electric forklift speed is 5km/h, the savings in hours will be:

$$\text{Total hours} = \frac{\text{kilometers}}{\text{km/h}} \tag{4.3}$$

- **1 year**: 230.2 hours
- **Week in June**: 3.9 hours
- **Week in December**: 4.3 hours

The results show the overall savings in hours for the respective time periods. 3.9 to 4.3 hours per week is not enough to reduce the workforce by a full-time worker. However, by utilizing part-time workers, the change would still yield financial improvements.
As seen in Fig. [2.5] the Aalborg DC is the smallest DC, which would imply smaller distances and shorter travel time in general.

When looking at the savings for an entire year, it can be seen that the two numbers for the respective weeks are slightly below average, if a year consists of 52 weeks. But nevertheless, a saving of 230.2 hours would equate to approximately 6 weeks' worth of work, considering an 8-hour per day worker, that could be saved on this single process.

As mentioned throughout this section, the calculations are built on assumptions. The flow at PNL is constituted of two different flows, namely PS and SS, with a multitude of sub-processes, such as truck arrivals and truckload information. But unfortunately, due to poor supply chain visibility and poor data quality, the calculations have several assumptions, that will be outlined in Section [4.1.4].

4.1.4 Assumptions

It is important to note, that different assumptions have been made in some of the equations throughout the section. When calculating equations like multiplying a distance with the dock door demand, this might simplify or ignore a complex system. This could yield results that do not accurately reflect the real world.
The reason for mentioning assumptions afterward is the fact that leaving such variables out will ultimately distance the results from reality making the quantification hard to rationalize since steps are missing from the process. Just to mention some of the things that have not been accounted for throughout the calculations are:

- On/off-loading time for the forklifts
- Turn rate of the forklift
- Other workers and hindrances
- The intrinsic randomness in operations
- Immediate dispatches

To understand the meaning behind the different assumptions that have been left out, each of them will be elaborated upon.

On/off-loading time for the forklifts

Forklifts are mainly used for one specific task which is to transport pallets in warehouse settings. In doing so, the on/off-loading process is a necessity to continuously lift and move pallets. This can be time-consuming and should be accounted for.

Turn rate of the forklift

The turn rate of a forklift is also something that should be considered. This is especially important when the setting is of a small scale since this will leave less space for the forklift to operate within. As already pointed out, the Aalborg DC is quite small compared to the other DCs Fig. [2.5]. Throughout the calculations, an estimated 5km/h has been set to reflect these turns which force the operator to both decelerate and accelerate more frequently. This average speed can vary, depending on things such as cargo, number of workers, and other obstructions. Obstructions are partly touched upon such as other workers and hindrances, thus this is not further commented on.

The intrinsic randomness in operations

In daily operations, different challenges arise and uncertainty is introduced into the system. Demand, operators, and disruption all vary from day to day, and this is challenging to account for in simple mathematical calculations.

Immediate dispatches

In the way the calculations are made, there is assumed to be no waiting time between the transportation of the cargo. This assumption in itself is questionable but necessary in terms of the initial calculation, but realistically, waiting time is a natural occurrence of almost every existing system, whether it is damage to a pallet or product, measuring pallet dimensions, technical issues (e.g. scanner defect), etc.

To better accommodate variations, such as randomness, forklift physics (e.g. acceleration and turns), and more accurate process estimates: such as time to load and unload, scan times, and work schedules; creating a more refined model with such capabilities is desired.

4.2 Simulation

As described in Section [4.1.2], it is possible to reduce the distance traveled and hence the time spent on operations by allocating to dock doors by demand groups. Furthermore, it was concluded that the quantification of the improvements includes assumptions such as the turn speed of forklifts, arrival rate, and other stochastic variations. Therefore this section will aim to address some of the limitations and provide a more representative quantification of improvements. Furthermore, it will be possible to test different scenarios to identify near-optimal solutions. This will be achievable through simulation, where possible scenarios can be tested and compared.

To understand why simulation as a tool is useful it is important to understand the stochastic nature of a system, such as a DC. Throughout the flow, there are many factors that vary through time. For example, it has been shown in Fig. [4.2], that the demand varies from day to day. Hence it is necessary to introduce this randomness to the model when simulating.

Simulation can be seen as a paradigm with the purpose of modeling and analyzing complex systems. The modeling of a system is defined as a "Simplified representation of a complex system with the goal of providing prediction of the system's performance measures (metrics) of interest." [Altıok and Melamed, 2010] The model's purpose is thus to allow the user to gain insights into the behavior of the system. However, it is important to state that the model is a simplification of the real system in order to keep complexity and the cost down. [Altıok and Melamed, 2010] The reasons to develop a simulation model can be;

- To evaluate the current system performance in different scenarios
- To predict the performance of experimental systems
- Testing different designs with the purpose of identifying trade-offs.

[Altıok and Melamed, 2010]

Models can be physical, mathematical, and computer-based, depending on the purpose. For example, a physical mini model of a Car could be used for wind tunnel testing, as it would be cheaper to construct compared to a full-scale version. A mathematical model such as the one used in Section [4.1.2] can be used to describe the workflow. [Altıok and Melamed, 2010] However a mathematical model is best suited if the model is relatively simple. Law [2015]
However, to test the system while presented with random variation this section will introduce a computer-based simulation model. [Altıok and Melamed, 2010]

Discrete Event Simualation Discrete Event Simulation (DES) is a simulation method for models that evolve over time, hence it is a sequence of events that change the model through time. [Law, 2015] As the name suggests, the model is controlled by discrete events that occur as time progress in the model. In a DC scenario, it could be the arrival of a truck with pallets that need to be sorted and distributed. [Altıok and Melamed, 2010]

The model has stochastic properties, meaning that the model will have some random input components. This approach is typically used in queuing and inventory systems. Thus, this approach seems appropriate for this study. Law [2015]

4.2.1 Model framework

There are many ways to construct a simulation model, and this thesis has followed the suggested framework by Altıok and Melamed [2010] for simulation modeling. The framework has eight steps, from analysis and data collection to validation and final recommendations based on model results:

1. 1. Problem Analysis and Information Collection
2. 2. Data Collection
3. 3. Model construction
4. 4. Model Verification
5. 5. Model Validation
6. 6. Designing and Conducting Experiments
7. 7. Output Analysis
8. 8. Final Recommendations

In the following sections, it will be described this thesis applies this framework to construct the simulation model and gather results with the purpose of recommending a solution.

4.2.2 1. Problem analysis and information collection:

This step aims to identify the problem, input parameters, performance parameters, relationship of the parameters, and the rules of the system operations. [Altıok and Melamed, 2010]

The problem has been identified as quantifying the distance reduction and overall improvements of dock door allocation in PNL's DK91 DC. Information and logic for the model: such as workforce capacity, dock doors in use, processes (e.g. how and where things arrive), and forklift network routing has been determined through interviews with PNL representatives.

4.2.3 2. Data Collection:

This step aims to gather data used as input parameters for the model. Based on this data, different assumptions and distributions can be implemented to reflect the data in the model. [Altıok and Melamed, 2010]
Most of the input data have been collected in Chapter [2], where data such as scan times, daily demand, and end destination have been collected. These data have been used to

define the arrival rate, which will be further described in Section [4.2.4]. Moreover, the authors have measured the dimensions of the warehouse facility to accurately map the facility. Other factors such as forklift's speed have been gathered in Section [2.3] from the manufacturer's websites.

4.2.4 3. Model construction

After the data and information have been collected the model can be constructed [Altıok and Melamed, 2010]. This section includes a selection of simulation software and a description of how the model of PNL's DK91 DC. **Software selection** As seen in Table [4.8], calculations indicate that PNL can reduce the cost of their DC operations by optimizing the dock door allocation. Therefore it was chosen to simulate different scenarios to identify and validate possible solution proposals. As a result Simulation software is needed. This will be selected based on the criteria below:

- Free to use
- Ability to create and simulate Discrete event simulation models.
- Support or learning material available

Based on these parameters, three tools were identified. These were; Simio, AnyLogic, and Enterprise Dynamics. These have been examined based on the parameters mentioned above with the purpose of selecting the best tool for the task. This can be seen in Table [4.9].

Provider	Free to use	Discrete event simulation	Support
Anylogic	Limited Student Access	Yes	Yes
Enterprice Dynamic	Uni License	Yes	Yes
Simio	Student Access	Yes	Yes

Table 4.9. Software Selection Table

As can be seen in Table [4.9], all examined tools have similar characteristics and features based on the selected parameters. This indicates that all tools will be suitable for the job. However, it was chosen to use Enterprise Dynamics, as Aalborg University has a collaboration agreement including support from Integrate providing learning resources and meetings. This provided the authors with relevant teaching material and necessary support.

4.2.4.1 Physical dimensions

As described in Section [4.2.3], the physical dimensions of the DC were measured, with the purpose of accurately calculating distances traveled by the transporters, and can be used for mapping the facility in the simulation model. The measures included the outer parameters of the building, storage places, dock doors, etc. This provided the authors with a blueprint of the facility.

The next step was to map the location of the Ground Storage Units at the facility. As PNL does not use racks, but rather places the pallets in marked areas on the floor, the *Ground Floor Atom* was used. Each *Ground Floor Atom* was then scaled in regard to pallet capacity, physical dimensions (x,y), and location in the facility based on the measurements performed by the group. Some dock doors have less capacity than others as a result of the layout of the facility. As can be seen on Fig. [4.1], there are two storage locations for most dock doors. This proved to be a challenge in the model as every *Ground Storage Atom* has its own channel and therefore does not function as one entity per dock door. Therefore it was chosen to place a third *Ground Storage Atom* between the two, with the purpose of distributing the demand to the *Ground Storage Atoms*s at the allocated dock door. This means that every pallet for the dock door will be delivered to the center storage atom of each dock door which then distributes the demand randomly to the two remaining *Ground Storage Atoms* at the dock door.

Lastly, the preliminary sorting had to be mapped. As the preliminary sorting, is a temporary storage where goods are stored across different dock door storage zones, it was chosen to represent the preliminary sorting in the model through *Queue* atoms. Through interviews with the company contact, the location of each queue was determined and the capacity was set to 1000. This was chosen as there are no finite capacity limitations, and thus the number was set well above the actual need of the queues. This ensures that the demand cannot ensure capacity.

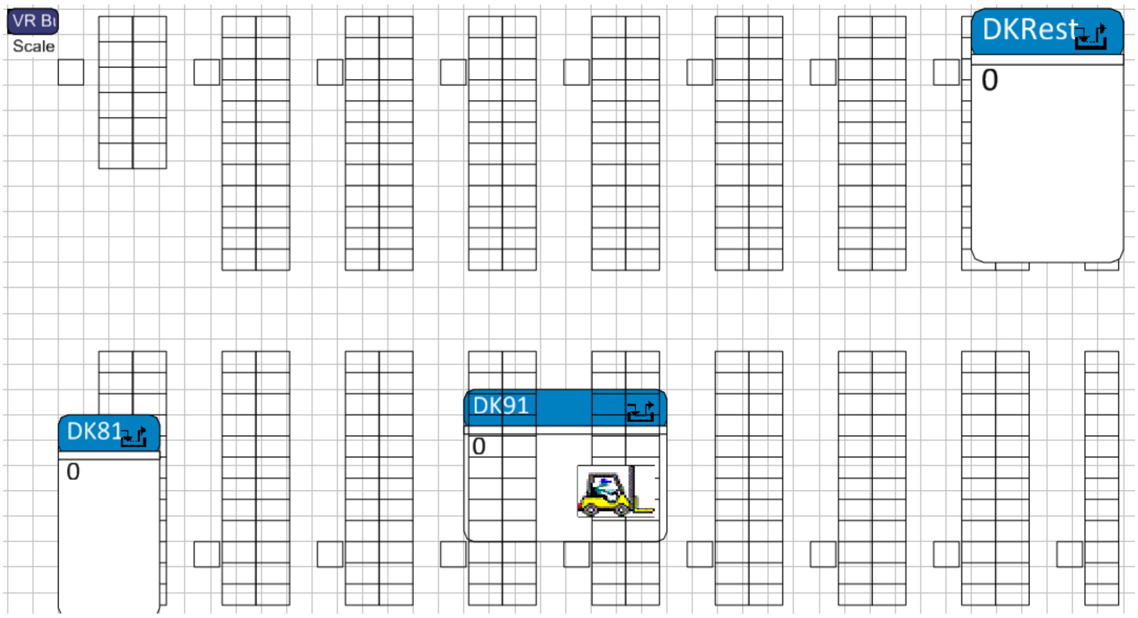


Figure 4.5. physical dimensions

Fig. [4.5], illustrates the results after mapping the physical dimensions in the model where every grid represents a square meter. The queues for the preliminary sorting are represented by the blue and white boxes whereas the dock doors are in between the *Ground Storage Atoms*. It was chosen not to map the storage units of the two westbound dock doors as they are not used for storage in the current flow. Therefore there is a deviation in comparison to Fig. [4.1].

4.2.4.2 Flow

This section seeks to describe how the flow of the facility was implemented in the model. The section will be divided into two parts: Arrival and process flow. The arrival section includes a description of how the arrival rate was determined and how the product flow arrives into the system.
The handling section describes how the pallets are moved around in the facility using electric forklifts.

4.2.4.3 Arrival

This section describes how arrival has been handled in the model. This mainly concerns flow, as truck arrivals could not be meaningfully modeled, an alternative by equally dividing demand across 6 dock doors for PS, and having a fixed off-load zone for SS in the western section of the DC. As a result, a *Source Atom* will be placed in the northern part of the

DC and another *Source Atom* in the western part of the DC

The first part of the flow is to define how demand arrives in the model. This has been replicated from the logic illustrated by Table [4.2] in Section [4.1].

Therefore it was chosen to utilize the KP2 data to define the arrival rate of goods. The data were filtered by the shipment and delivery department to separate the goods that arrive from line hauls from other PNL DCs and the goods that arrive through local pickups.

As illustrated in Fig. [4.1] demand arrives from different dock doors. Where the goods arrive, depends on the geographical origin of the goods. Therefore the decision parameters presented in Table [4.2] have been utilized to filter demand.

To implement this logic into the model it was chosen to use the *Arrival List* atom. As can be seen on Fig. [4.6], a sample of the data has been shown. The Arrival list contains four critical parameters, which are described below:

- *Arrival Time*: hr(x) specifies arrival time in hours.
- *Quantity*: The quantity that arrives, at the specified time
- *Dest*: A label that determines which dock door the demand should be delivered to.
- *Grovdest*: A label that determines what/ if any preliminary queue the demand should be delivered to.

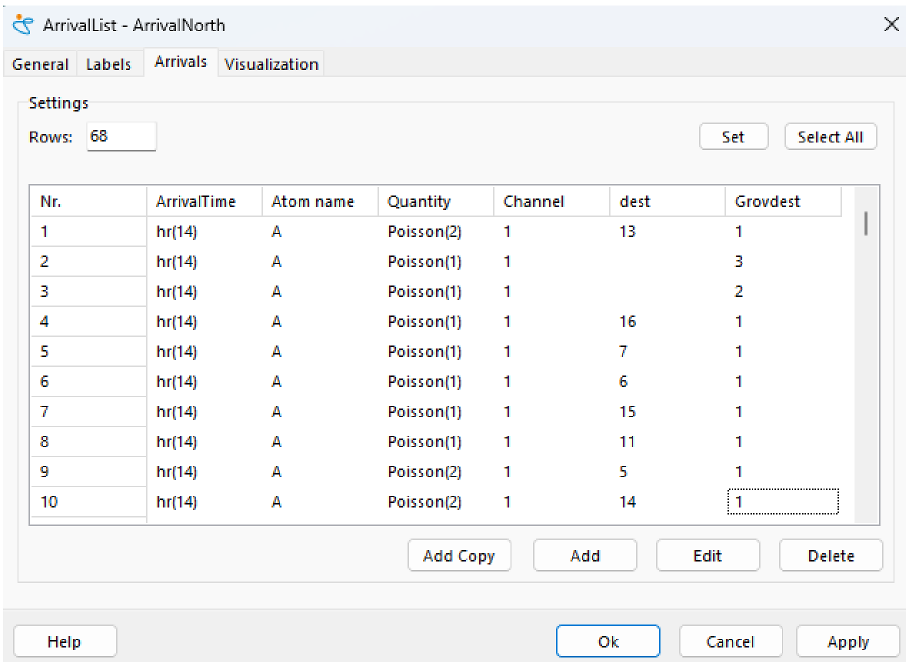


Figure 4.6. Preliminary Sorting Arrival List

It was chosen to specify arrivals hourly, as the data for the KP2 scans are given in time stamps down to the second; these were resampled into hourly brackets. However, this also means that all demand for one hour is introduced to the model at the exact same time. However, to keep model complexity down, it was deemed as sufficient as the exact arrival time is unknown. To introduce a stochastic element of randomness into the arrival pattern in the quantity parameter. Thus, it was chosen to use a stochastic distribution for the arrival quantity. It was chosen to use a Poisson distribution as it is a discrete probabilistic arrival function. Meaning that it will output integer values based on the inserted mean value, a common practice for arrivals in discrete-event simulation. [Sharma et al., 2021]

Figure 4.7, illustrates the probability of different value outputs based on a mean value of three. Here, number of discrete occurrences is distributed according to the mean, with the probabilities gravitating towards the mean.

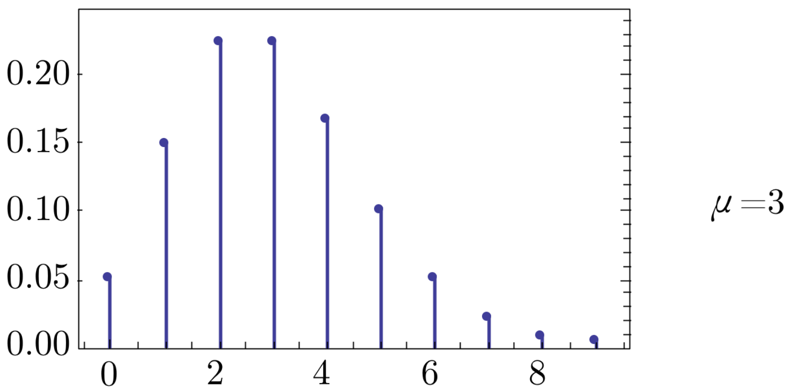


Figure 4.7. Example of a Poisson Distribution with a Mean of 3

As all the demand for each hour will spawn simultaneously in the first second of the hour, it was chosen to use queues to absorb the demand at the dock doors. An illustration of how this works in the PS can be seen in Fig. [4.8]. The figure illustrates that the *Arrival List* feeds a Distribution Queue, which then distributes the demand equal to the different dock doors' unique queues to reflect trucks that arrive. This was done through the following Bernoulli function:

- e1 = Probability value e2 outcome
- e2 = True value
- e3 = False Value
- e4 = specified random generator, if none are selected 1 will be used by default.

Thus, four values are accepted by the function (where only three are used), and to achieve the desired functionality, we nest the function to render multiple outcomes:

$$\begin{aligned} & Bernoulli(16.66667, 1, Bernoulli(20, 2, \\ & Bernoulli(25, 3, Bernoulli(33.333333, 4, \\ & Bernoulli(50, 5, 6))))) \end{aligned} \qquad [4.4]$$

The Bernoulli function states the probability of a specified outcome. However, as the demand needs to be distributed across six dock doors it means that the function has to be nested. Hence the first Bernoulli is the probability of dock door 1. The next Bernoulli function then needs a new percentage where the previous demand is subtracted from the total. This process is then repeated until every dock door has been covered.

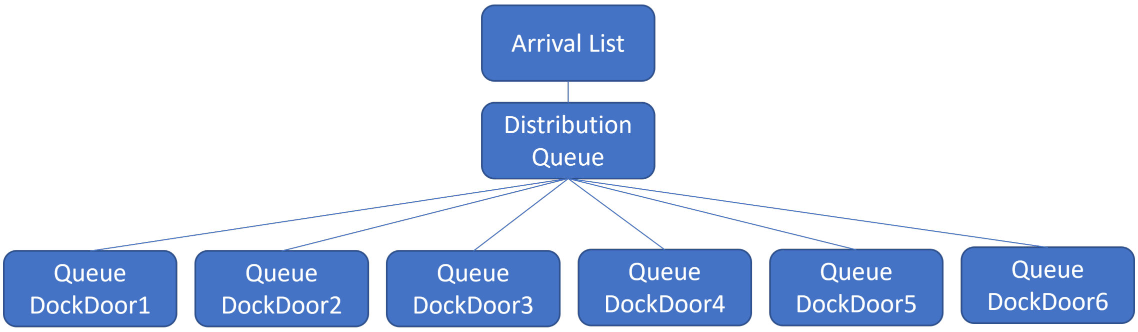


Figure 4.8. Arrival Distribution Chart

The process was then repeated for the goods that arrive through the west part of the facility. However as seen on Fig. [4.1], there are only two dock doors for arrivals. These are special as they have a hydraulic dock to enable trailers with no lift to dock.

Creating the data for the Arrival list can be tedious to do manually as it has to be sorted by hour and each dock door's demand, for every hour, needs to be specified on a separate line.

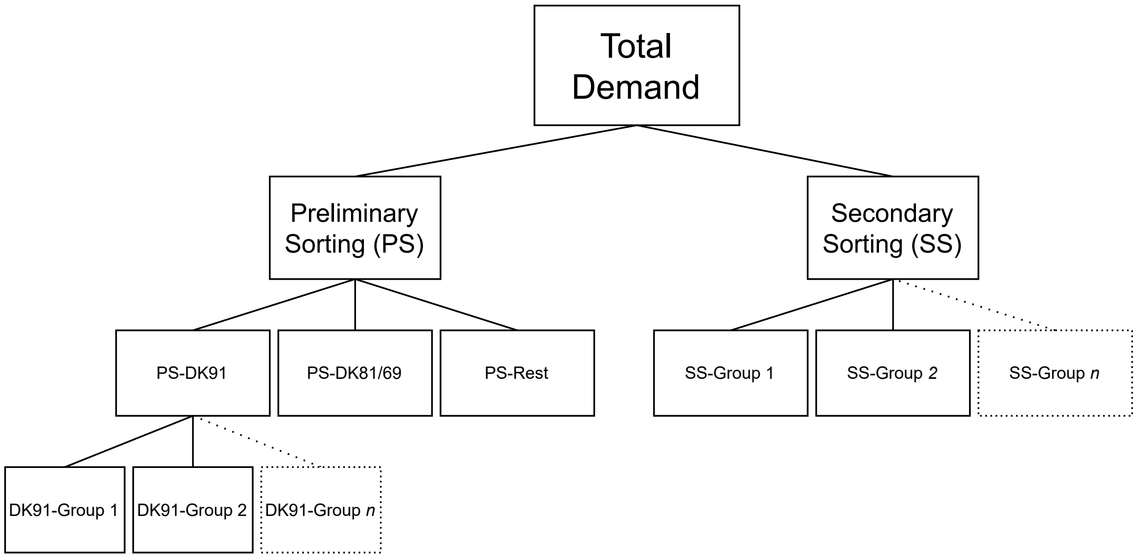


Figure 4.9. Demand Structure for PNL (DK91)

As seen in Fig. [4.9], starting with preliminary sorting, all outgoing demand to DK91 (that has to be pre-sorted), is split into the appropriate demand groups as provided in Section [4.1]. This equals 13 demand groups with DK81/DK69, Aarhus, and Herning, and the remaining demand represents their own distinct group. Thus, 15 groups have been created for preliminary sorting PS-DK91 has 13 subgroups with additional groups from PS-DK81/69 and PS-Rest, thus representing all preliminary sorting demand. For secondary sorting, the demand is simply split into the same demand groups as PS-DK91. This equates to 13 additional groups totaling 28 demand groups to account for, that has to be resampled into 24-hour bins. This has been achieved in Python and the source code can be found in Appendix [E.2].

Hence to conclude on the arrival in the simulation model. Arrival lists are used to spawn demand, the demand is then sent to a Distribution queue where demand is picked up for the secondary sorting.

4.2.4.4 Handling

The next step is to configure the transportation network. For a transportation network to function as intended for this thesis the following Atoms are needed:

- **Advanced Transporter**: The electric pallet lifter
- **Network Nodes**: Atoms that can be placed in the facility to mark the direction and turning points of a network.

- **Node manipulator**: Atom that allows the *Network Nodes* to be connected to other atoms.
- **Network controller**: A Controller that calculates the shortest path between network nodes.
- **Dispatcher**: Connects to the Atoms where goods are picked up by the advanced transporter.
- **Destinator**: Connects to the Advanced Transporter and the destinations.
- **Availability control**: Connects to the Advanced transporters and Time Schedule Availability.
- **Time schedule Availability**: Connects to the availability control and defines the operating hours.

The first step is to build the routing network that the transporters will use. Through observations, it was found that the workers did not necessarily transport the goods on specific transportation lanes. However, this would be challenging to simulate as the floor is gradually filled with pallets over time. Furthermore, it could be seen as a safety hazard as transportation is random across the floor. Therefore it was chosen to represent the transportation of electric forklifts on a pre-defined routing network, based on floor marks in the facility. The network was built by placing *Nodes* at every turn and connecting them with the *Node Manipulator*. Furthermore, each *Ground Storage Atoms* and *Arrival Queue* were connected to the path using the *Node Manipulator*. It was chosen to make all the paths bidirectional as there is currently no lane separation for transpiration in either direction. The last step is to place a *Network Controller* in the model and enable the '*Optimize network on reset*' option, to ensure that the shortest paths are calculated after every reset. A depiction of the network can be seen on Fig. [4.10].

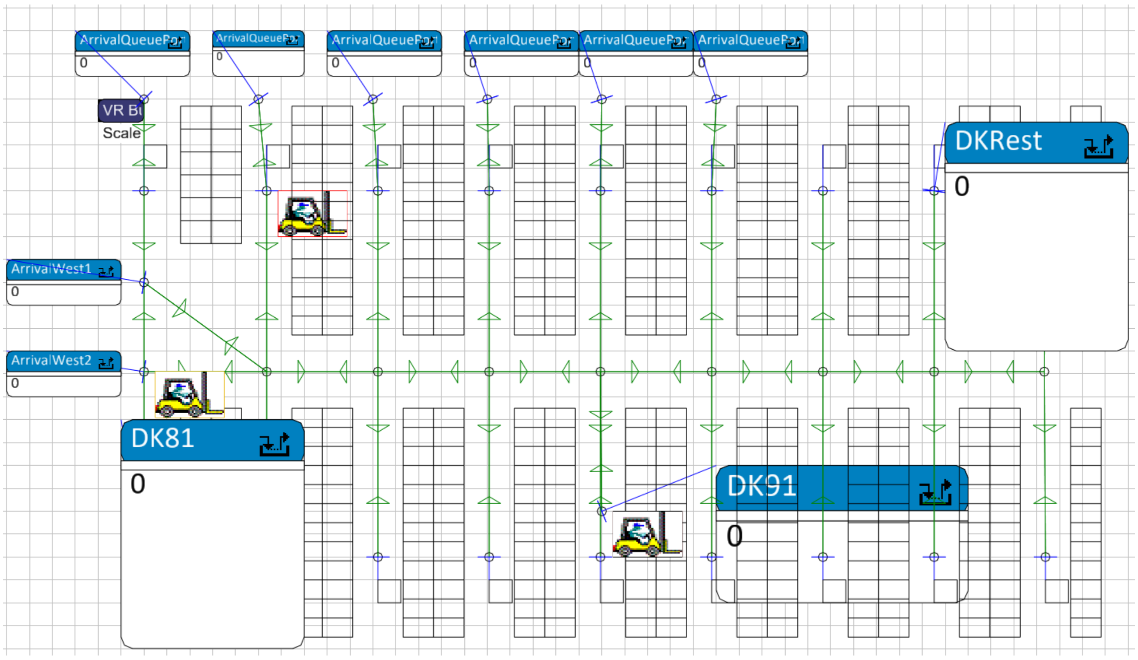


Figure 4.10. Illustration of the network in the model

The next step is to add the electric forklifts to the model. As seen on Fig. [4.11], there are several connections between atoms, to ensure that the transporters function as intended. All connections to the atom are managed in the yellow boxes. The left side function as input to the atom, the middle (yellow) as a control/information input, and the right side as output. For the flow depicted in Fig. [4.11], the *Dispatcher* input is the arrivals from the West and the DK91 queue. The output is connected as an input to the *Advanced Transporter* atoms.

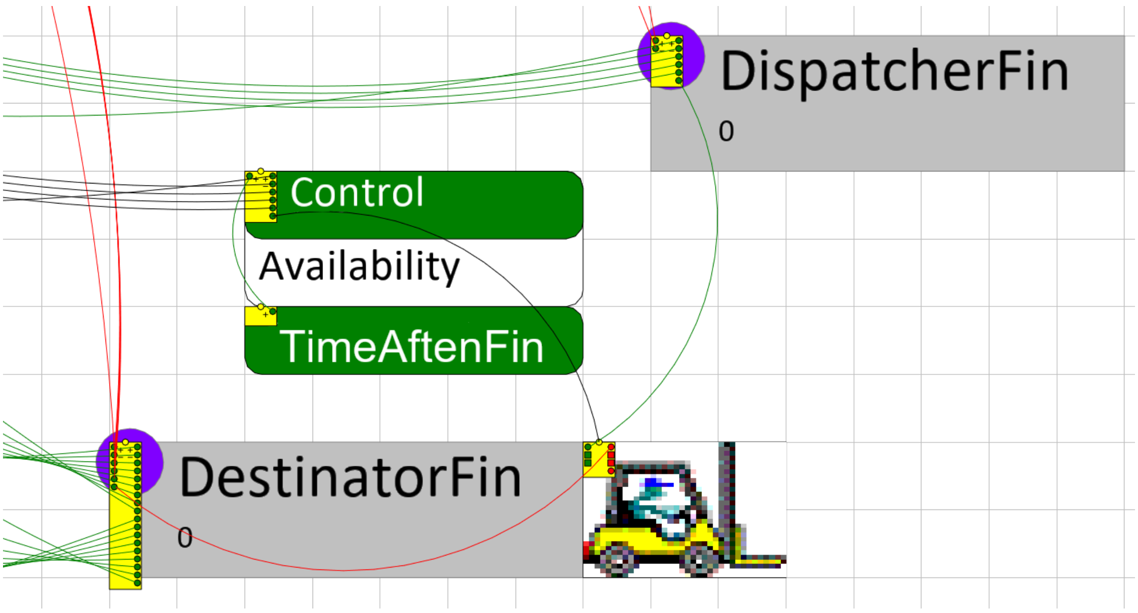


Figure 4.11. Connections to an Advanced Transporter in Preliminary Sorting

To control the operating hours of the advanced transporter a *Time Schedule Availability* atom is connected to a control atom which is then connected to the advanced transporters. This allows the modeler to determine the up and down time. The output of the *Advanced Transporter* is connected to the *Destinator* atom which has the output for every dock door.

It is important that the output of the *Destinator* atom is applied in numeric order as the advanced transporters will read the 'dest' label in the arrival list as the channel number of the Destinator. Hence label = 1 will be delivered to the first atom that is connected to the Destinator output.

With the flow in place, the next step is to configure the advanced transporters. This step is similar for both the PS and SS and therefore they will both be described based on the same illustration.

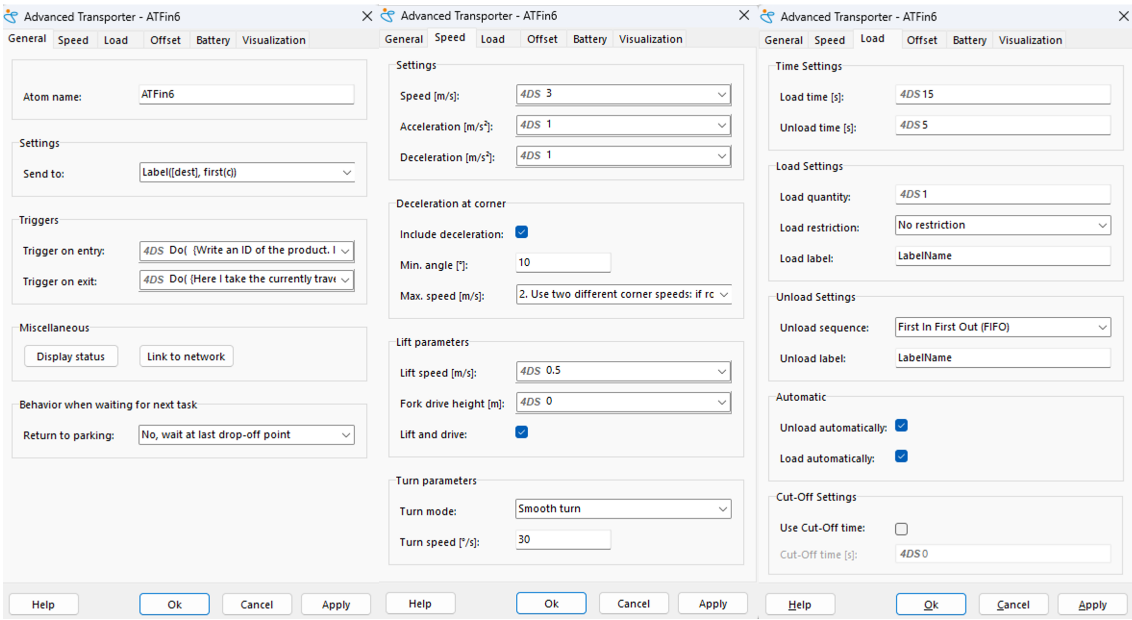


Figure 4.12. Configuration of a Secondary Sorting Advanced Transporter

Fig. [4.12], illustrates all the parameters that have been changed for the simulation model. All parameters are the same except for the 'Send To' and naming.

Naming of the *Atom name* object is based on the purpose and number of the process to easily identify each *Advanced Transporter*. The *Send to* is a key parameter as this is where the destination of each pallet is defined. Through the use of 4D Script, it is defined that the secondary sorting dock door destination should be based on the label 'dest' for every row. Hence the only difference for PS is changing 'dest' to 'grovdest'. The *Triggers* are not critical for the model to function as they are solely used for testing distances in the

model. The *Return to parking* was set to 'no' as the model does not include the parking area or charging station.
The *Speed* column was defined based on manufacturers' specifications. Where it was found that the forklifts have a top speed of 3 m/s. The remaining parameters were left default. In the *Load* column the *Load and Unload Time* were specified based on estimates. Through interviews, it was found that the workers need to scan the product and identify the postal code prior to moving the pallet.
It was found that 15 seconds were representative of the loading process including physically lifting the product, scanning the product, and identifying the postal code. Likewise, it was chosen to include a unload time of 5 seconds. The remaining parameters were left default. The only difference for the PS transporters is the *Send to* parameter in which 'dest' is substituted with 'Grovdest'.

Hence this flow will ensure that goods that have been sorted for DK91, and goods that arrived through linehaul from other DC's are picked up and moved to the correct dock door.

Through trial runs it was found that the daily demand per dock door could exceed the 22 pallet capacity. Through interviews, it was found that multiple routes could be outbound from the same dock door. To solve this issue it was determined to continuously drain the dock doors for products by connecting them to a sink. This compromise was found acceptable, as the model stalls if it has to deliver products to a filled dock door. However, it also means that at one given time in between shipments more than 22 collies could potentially be at a dock door. However, as there was no data for outbound shipments this was found to be an acceptable compromise.

This is the model configuration that will be used to test potential improvements and the As-Is state.

With all of these atoms, settings, and measures in place the model is ready for verification, and results gathering afterward.

4.2.5 4-5 Model verification and validation

According to Robinson [1997] verification and validation (V&V) are seen as a way to ensure the model is constructed in accordance with its real-world counterpart and should be evaluated jointly.

V&V is seen as a method to increase confidence in a simulation model's results, as no model is an exact replica of reality. Verification is a process that ensures that the real concept has been computerized with sufficient accuracy. Whereas, validation is the process that ensures, that the build model suits the right purpose, meaning that it can investigate the problem at hand. Sufficient accuracy, according to [Robinson, 1997], is relative to the intended purpose of the model, in other words: whether the model achieves and reflects the desired goal.

Since each step throughout the building and evaluation process of building a simulation requires parallel verification and validation, the two terms will be used to coincide. The four different steps used for V&V will be listed and further elaborated upon below. [Robinson, 1997]

4.2.5.1 Conceptual Model Validation

In order to achieve conceptual model validation, it is necessary to gain in-depth knowledge about the system that needs to be modelled. This needs to be done in collaboration with a system expert. Through the conceptual model validation, the model used in this thesis has been compared to the real system, in collaboration with the PNL representative. By doing so, cohesion between the model and the real world was achieved. [Robinson, 1997]

4.2.5.2 Data Validation

The data for the model has been validated within the possible means. By this, the source of the data i.e. the representative has been used as a reference point for accurate the data was in terms of reflecting reality. Besides, the data has been stored separately to allow for easier error detection, when the model was not performing as expected.[Robinson, 1997]

4.2.5.3 Verification and White-Box validation

White-Box V&V as a term, pertains to ensuring that the model was true to the conceptual model. During the White-Box V&V, various aspects have been considered in order to ensure that the content of the model reflected the real world. In order to achieve this, the aspects that can be used for this purpose are presented as:

- **Travel times**; E.g. the travel times and speed of the forklifts and how they manoeuvre as they operate. Eq. [4.2]

- **Shift patterns**; As mentioned in Section [2.3.2] the operations are split into two different time intervals (preliminary and secondary working hours). This aspect has also been accounted for in the model.
- **Routing**; In order to accurately represent the routing, the forklifts have been assigned to a network, meaning that they can only travel alongside the lines depicted on the model overview. This ensures cohesion between the model and reality. Fig. [4.10]
- **Distribution sampling**; as described in Section [4.2.4.3] a distribution has been sampled from empirical data. This represents the arrival of cargo, from a probabilistic perspective. [Robinson, 1997]

Apart from the listed aspects, another thing that has been used to conduct White-Box validation is the cross-validation of output reports. During the life-cycle of the simulation, a multitude of outputs has been generated and evaluated as the model progressed. The different problems encountered in this regard will be further reflected upon in Chapter [7].

4.2.5.4 Black-box validation

For this validation, the results of the simulations can be compared to the ones gathered from the real system. This comparison has not yet been done, as the results for the simulations are yet to be presented. Still, this point is included since it is an imperative step in the framework. Another possibility is to compare the model to alternative models, but this is primarily used when real data is lacking. Since this is not the case for this specific problem, this approach has not been implemented.[Robinson, 1997]

4.2.6 6. Designing and conducting simulation experiments

This section introduces how the simulation experiments were conducted, with the purpose of collecting relevant data.

When running any given simulation, most practitioners are introducing warm-up time before each run. The warm-up time is needed in order to allow the stochastic process to be present in the system, meaning that it reaches a steady state.[BIRTA, 2021] But, since the model presented for this analysis instantly generates a probabilistic amount of entities to feed into the model, the warm-up period was deemed unnecessary. Despite this, a test was performed on a 15-hour warm-up time, to test the impact of starting the model with the preliminary sorting team. However, it was found that the data from the warm-up period was included in the results and thus this method was disregarded.

The first parameter that needs to be determined is the confidence level. The confidence level in Enterprise Dynamics can be set to either 80%, 85%, 90%, 95%, or 99%. Each level defines the percentage of outliers that will be removed. E.g. a 95% confidence interval means 2.5% will be removed on either side of the distribution. Each level tells us how confident each run will be in representing the true population mean of the data, as the experiments are run repeatedly [Witte and Witte, 2019]. For the simulations run, it has been decided to implement a 95% confidence level, as this is the most commonly used among practitioners. This is because the 95% confidence level balances the level of certainty together with marginal error, in such a way, that it is applicable to most scenarios [Cumming, 2013].

Another concern that needs to be determined when running a simulation, is the number of iterations. The selection of iterations can be challenging and mathematically comprehensive, yet it is necessary since the number of iterations influences the accuracy of the results. [BIRTA, 2021]
If too few repetitions are selected, accuracy and precision are lost and results may be distorted; conversely, if too many repetitions are selected, valuable time is wasted. [Hoad et al., 2010]
By performing a set number of repetitions, a set of independent and identically distributed output variables is created. Over time, with many repetitions, the results should be approximately normally distributed, as determined by the central limit theorem. In other words, as the number of replications gets larger, the distribution of the averaged results converges to normality. [Hoad et al., 2010].

To obtain the number of required runs to achieve this, a number of repetitions were tested to ensure normality convergence:

	Repeptitions				
Features	**50**	**100**	**200**	**400**	**1000**
LB (2.5%)	566.19	579.28	582.58	582.96	584.42
Avg	576.10	586.21	587.53	586.17	586.54
UB (97.5%)	586.01	593.14	592.47	589.38	588.66

Table 4.10. Comparison of Number of Runs and Arrival Statistics (February 2022 - February 2023

In Table [4.10], statistical features from a range of repetitions performed on demand arrivals have been provided. It can be seen, that the lower bound (LB) and upper bound (UB) values converge towards the mean, reducing variance with more repetitions. From 50 repetitions the results are normally distributed, but the variance is high. As repetitions increase the variance within the 95% is reduced to ≈ 2 in 1,000 repetitions from ≈ 10 in 50 repetitions.

From these tests, it was chosen to use 1000 runs as it provided a good foundation. However, more runs could reduce variance even further, and render the results statistically better.

The logic behind the door allocation dock door allocation is similar to the logic that was followed in Section [4.1.2], where the largest demand is matched with the inbound dock door closest to its final destination. Furthermore, it was decided to not utilize the two outermost left dock doors as they only have a 25% capacity in comparison to the remaining dock doors, as depicted in Fig. [4.1]. Therefore the dock allocation for scenarios 2 and 3 will be as follows: [14, 19, 13, 10, 12, 21, 11, 22, 10, 23, 9, 24, 8]. However as mentioned in Section [4.2.4], the 'Dest' and 'Grovdest' labels are based on sequence from 1-x dock doors and thus the model utilizes other values in the illustrations in Section [4.2.4].

The following three scenarios will be tested across 3 periods, thus a total of 9 experiments will be made.

Scenario	**Experiment**	**Start Date**	**End Date**	**PS As-Is**	**Allocation As-Is**
1	1	01-feb-22	01-feb-23	✘	✘
2	2	01-feb-22	01-feb-23	✘	
3	3	01-feb-22	01-feb-23		
1	4	01-aug-22	31-aug-22	✘	✘
2	5	01-aug-22	31-aug-22	✘	
3	6	01-aug-22	31-aug-22		
1	7	01-feb-23	31-feb-23	✘	✘
2	8	01-feb-23	31-feb-23	✘	
3	9	01-feb-23	31-feb-23		

Table 4.11. Scenario List

It was chosen to test a larger dataset of a year in comparison to two monthly datasets to identify if the time period impacts variations in demand and results. Furthermore, it was chosen to run a monthly dataset from February and August to test two different seasons.

For scenario 3, the DK91 PS Zone Fig. [4.1] that is used in the As-Is operations and Scenario two has been relocated, which is depicted in Fig. [4.13]. This is implemented to further reduce PS travel distance, by moving the zone closer to the six inbound dock doors.

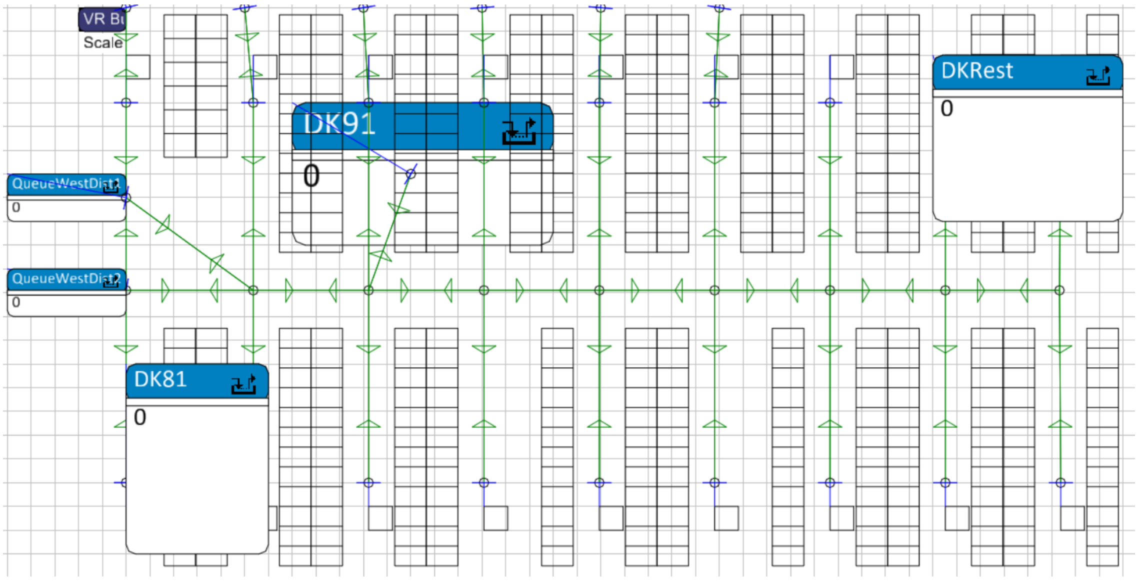


Figure 4.13. Simulation Setup for Scenarios 3, 6, and 9

To measure the model performance it was chosen to focus on the following KPIs:

- Average Total Distance
- Average Distance Preliminary Sorting
- Average Distance Secondary Sorting
- Average Utilization of Advanced Transporters

These KPIs were chosen as these relate to improving the PNL Scan Throughput KPI, as to reduce the resources spent on the transportation of goods.
By having static loading and offloading times, the model also tests this through the utilization parameter, where it would be expected that the utilization would decrease if the efficiency is increased resulting in better use of resources.

Multiple test runs were performed to ensure that the model performed as intended with live on-screen measurements such as the *Generic Monitor* that allows the modeler to monitor the output to each dock door during the simulation. Furthermore, tables were used to review distances live. This was used to validate that distributions worked as intended.

However, this method is not ideal to monitor results across many runs, as it will involve manually note the results for every run. Therefore it was chosen to use the *Experiment Wizard* function in Enterprise Dynamics. The *Experiment Wizard* allows the modeler to collect data across multiple runs.

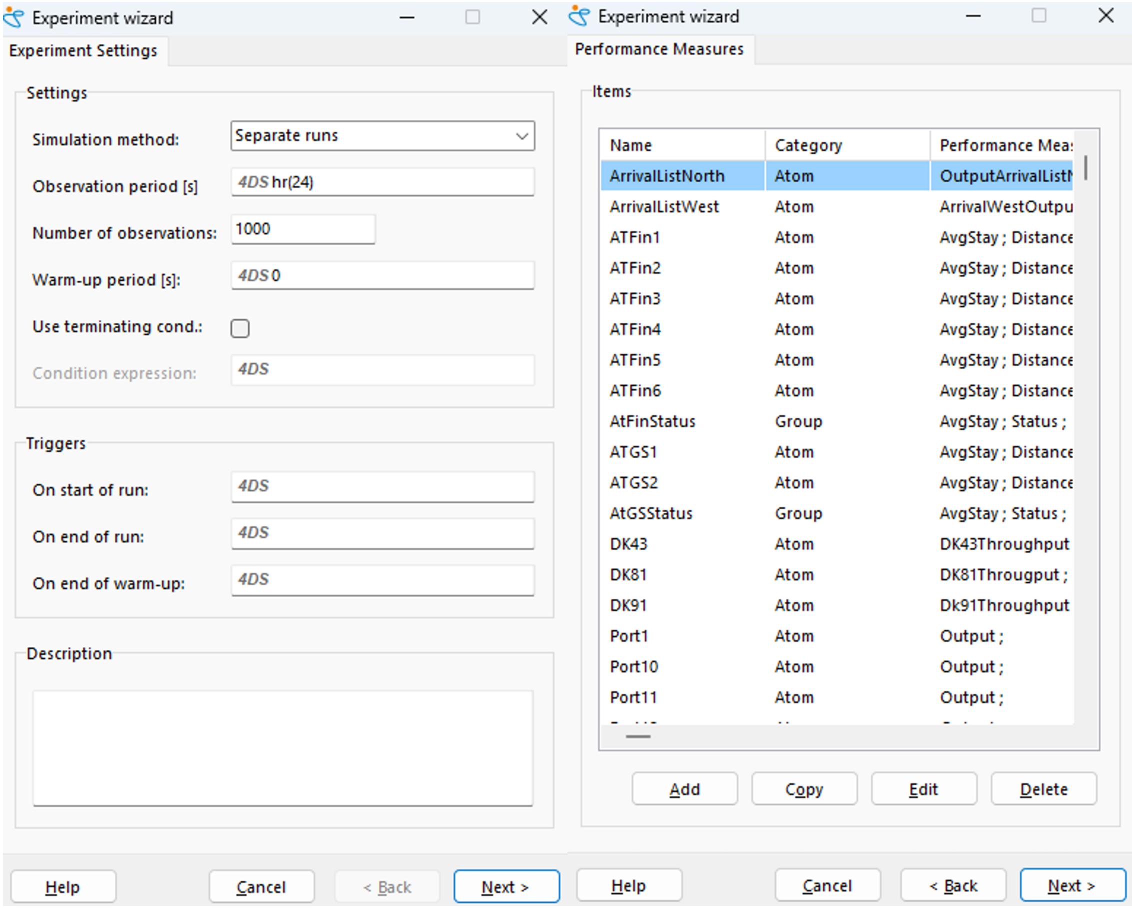


Figure 4.14. Configuration of the Experiment Wizard

Fig. [4.14] illustrates the configuration that was made for the data collection of the different simulation runs. Every Item has a distinct 4D script code to configure the measures. For example, the following measures were used for the Advanced Transporter:

AvgStay = AvgStay(CS)

Distance = Att([distancetraveled], cs)

Status = [Status]

The second screenshot in Fig. [4.14] illustrates the list of measures that were used. This includes monitoring the output at every dock door, the distances traveled by each advanced transporter, the status of the advanced transporters grouped as PS and SS, and lastly the input and output of the model. With this in place, the next step is to analyze the results.

4.2.7 7. Output analysis

This section presents the experiment results performed in the *Experiment Wizard*. The results have been imported to Excel to summarize distances and utilization of the advanced transporters, sorted by the process. Thus, this section will quantify the reduction of distance and labor achieved by the simulation scenarios.

As described in Section [4.2.6], the following measures were chosen to reflect and analyze the results:

- Average Total Distance
- Average Distance Preliminary Sorting
- Average Distance Secondary Sorting
- Average Utilization of Advanced Transporters

Through test runs it was found that the utilization was low, therefore it was determined to conduct the experiments with a half workforce to compare the results, as a potential for improved efficiency, would be to alter the workforce, if it does not hinder performance (i.e. the same work can be achieved with fewer workers). This is also tested, as the actual workforce on a given day is unknown.

Table [4.12], illustrates the total distance driven per 24 hours. As can be seen, both scenarios 2 and 3 reduce the distance. However, Scenario 3 consistently performs the best across all experiments. Furthermore, it can be seen that the reduction from scenarios 1-2 is greater than the reduction from 2-3. This was expected, as the allocation of dock doors affects every DK91 pallet whereas the placement of the DK91 PS zone only impacts $\approx 3.4\%$ of total demand.

Scenario	Start Date	End Date	1-3 Workforce	2-6 Workforce
1	01-feb-22	01-feb-22	40,378.18	40,713.25
2	01-feb-22	01-feb-22	36,912.64	37,200.23
3	01-feb-22	01-feb-22	**36,592.36**	**36,837.66**
1	01-aug-22	01-aug-22	33,764.90	33,570.90
2	01-aug-22	01-aug-22	30,070.55	30,139.86
3	01-aug-22	01-aug-22	**29,819.60**	**29,681.61**
1	01-feb-23	01-feb-23	32,615.18	32,557.26
2	01-feb-23	01-feb-23	29,581.01	29,470.04
3	01-feb-23	01-feb-23	**28,936.78**	**29,342.00**

Table 4.12. Total Average Distance in meters per Day

To understand how these improvements are attained across the PS and SS teams, the results for each shift are presented.

			PS		SS	
Scenario	**Start Date**	**End Date**	**1-3 Workforce**	**2-6 Workfroce**	**1-3 Workforce**	**2-6 Workfroce**
1	01-feb-22	01-feb-22	15,248.04	15,207.73	25,130.15	25,505.52
2	01-feb-22	01-feb-22	15,249.73	15,253.40	**21,662.91**	**21,946.83**
3	01-feb-22	01-feb-22	**14,846.75**	**14,764.68**	21,745.61	22,072.97
1	01-aug-22	01-aug-22	9,748.28	9,744.35	24,016.61	23,826.54
2	01-aug-22	01-aug-22	9,749.27	9,793.46	**20,321.28**	20,346.40
3	01-aug-22	01-aug-22	**9,456.19**	**9,414.77**	20,363.42	**20,266.85**
1	01-feb-23	01-feb-23	11,975.43	11,974.70	20,639.75	20,582.56
2	01-feb-23	01-feb-23	12,042.42	11,972.72	**17,538.59**	**17,497.31**
3	01-feb-23	01-feb-23	**11,385.51**	**11,711.26**	17,551.27	17,630.74

Table 4.13. Total Daily Distance in meters for PS and SS

Table [4.13], illustrates the distances driven by each team. For the PS team, it can be seen that scenario 2 does not reduce the distance driven for these workers. This is to be expected as the DK91 location remains in the original position across both scenarios. It would therefore be expected that the results were identical. However, a small variation can be seen in the results. This could be caused by the randomness of the model. However, the distance is reduced for scenario 3 where the DK91 PS zone is moved in between the 6 arrival docks. Although it must be noted that the effect of this change is not as large as the allocation of dock doors.

For the SS team, it can be seen that distances are reduced in Scenario 2. This is expected as the dock door allocation changes. It can further be seen that Scenario 2 performs slightly better for the SS team. Furthermore, it can be seen that the difference in distance between scenarios 2 and 3 is not as large. This was also expected as the DK91 PS area only constitutes $\approx 3.4\%$ of the total DK91 demand. Thus there are fewer pallets that are affected by the change.

The next step is to examine the scenario's impact on the utilization. The utilization is calculated based on values from the *STATUS* 4D script. Where the following values are presented:

- Status Idle (SI)
- Status TravelFull (STF)
- Status TravelEmpty (STE)
- Status Not Available (SNA)
- Status Load (SL)
- Unload (U)

The STATUS measure measures all stages of the model including the time when the advanced transporters are not available. Therefore the following calculation was used to isolate the utilization of the time available.

$$\text{Utilization} = \frac{\text{STF} + \text{STE} + \text{SL} + \text{U}}{\text{SI} + \text{STF} + \text{STF} + \text{STE} + \text{SL} + \text{U}} \quad [4.5]$$

Scenario	Start Date	End Date	1-3 Workforce	2-6 Workfroce
1	01-feb-22	01-feb-22	27.89%	14.12%
2	01-feb-22	01-feb-22	27.13%	13.70%
3	01-feb-22	01-feb-22	**27.07%**	**13.66%**
1	01-aug-22	01-aug-22	23.09%	11.21%
2	01-aug-22	01-aug-22	22.12%	**10.80%**
3	01-aug-22	01-aug-22	**22.10%**	10.84%
1	01-feb-23	01-feb-23	22.39%	11.51%
2	01-feb-23	01-feb-23	21.65%	11.11%
3	01-feb-23	01-feb-23	**21.33%**	**11.01%**

Table 4.14. Total Utilization of All *Advanced Transporters* per Day

Table [4.14], illustrates the average utilization of the *Advanced Transporters*. However, as described the two workforces consist of a 1-3 worker setup and a 1-6 worker setup. Therefore the average values have to be weighted based on the number of workers that participate in the work at the given shifts. Therefore the average values are calculated as follows:

- U_{PS} = Utilization Preliminary Sorting
- U_{SS} = Utilization Secondary Sorting
- NW_{PS} = No. of Workers Preliminary Sorting
- NW_{SS} = No. of workers Secondary Sorting

$$\frac{(U_{\text{PS}} \cdot NW_{\text{PS}}) + (U_{\text{SS}} \cdot NW_{\text{SS}})}{NW_{\text{PS}} + NW_{\text{SS}}}$$

This ensures that the SS values are weighted greater as they have a greater influence on the total time spent. The measure illustrates the percentage of time that the *Advanced Transporters* are operating, in relation to the total time available. Thus it is desirable to have a lower value as it means that the same workload has been handled in less time.

It can be seen in Table [4.14] that the utilization of the 1-3 workforce is half the utilization of the 2-6 workforce experiments. These results were expected as the workload is the same, and the workforce capacity is half the size. It can be seen that the values in scenarios 2 and 3 are close in proximity. Therefore random variation might impact the results, as they have to be differentiated on the third decimal. By averaging and comparing the results of each scenario, a total of 3.92% can be reduced for the 1-3 workers set up. However, this includes a static 20-second load and unloading process. Thus the actual transportation time reduction is greater.

To further understand the impact of every scenario, the utilization is disaggregated to the PS and SS levels.

			PS		SS	
Scenario	**Start Date**	End Date	**1-3 Workforce**	**2-6 Workfroce**	**1-3 Workforce**	**2-6 Workfroce**
1	01-feb-22	01-feb-22	42.885%	21.429%	22.893%	11.683%
2	01-feb-22	01-feb-22	42.887%	21.489%	21.877%	**11.109%**
3	01-feb-22	01-feb-22	**42.677%**	**21.279%**	**21.861%**	11.115%
1	01-aug-22	01-aug-22	27.547%	13.791%	21.610%	10.749%
2	01-aug-22	01-aug-22	27.536%	13.866%	20.320%	10.189%
3	01-aug-22	01-aug-22	**27.351%**	**13.638%**	**20.355%**	**10.139%**
1	01-feb-23	01-feb-23	33.335%	**16.713%**	18.741%	9.376%
2	01-feb-23	01-feb-23	**33.525%**	16.707%	**17.691%**	**8.827%**
3	01-feb-23	01-feb-23	**32.220%**	**16.644%**	17.696%	8.901%

Table 4.15. Utilization of the PS and SS of *Advanced Transporters* per Day

As seen on Table [4.15], scenario 3 generally has the lowest values. However as seen in experiment 1, As-Is performs the best for 1-3 workforce, with values that are close in proximity. This is a result of the static loading and unloading time as they constitute 20 seconds of every run. The status measure also shows the average time in the system per product, which varies from approximately 18.5-20.5 seconds. Thus half of the time is spent on loading and unloading.

Lastly, the system was monitored for pallets stuck in the system. As can be seen in Table [4.16], there are multiple pallets stuck in the system for the 1-3 workforce team for the yearly demand. It was identified that the Pallets that were stuck in the system, are left in the DK91 PS area. This is not ideal for a simulation model, as it means that the SS distance of those pallets is not accounted for and thus distorts the results.

Scenario	**Start Date**	**End Date**	**1-3 Workforce**	**2-6 Workfroce**
1	01-feb-22	01-feb-22	11.47	0.94
2	01-feb-22	01-feb-22	12.41	0.42
3	01-feb-22	01-feb-22	8.90	0.55
1	01-aug-22	01-aug-22	0.07	0.00
2	01-aug-22	01-aug-22	0.01	0.00
3	01-aug-22	01-aug-22	0.01	0.00
1	01-feb-23	01-feb-23	0.02	0.00
2	01-feb-23	01-feb-23	0.00	0.00
3	01-feb-23	01-feb-23	0.00	0.00

Table 4.16. Pallets Stuck In DK91 PS

This is a result of the time setup of the model. The model starts at 00:00 and ends at 24:00. The PS workers start their shift at 23:00 and thus there is an overlap in their working hours. Therefore they only have one hour in the model to empty the demand from DK91 PS, whereas in reality, they have time from 23:00-8:00. Thus it is assumed that the workers would be able to handle the excess demand throughout the SS shift.

It is therefore not ideal for the results but is evaluated as a non-significant challenge for implementation in real life.

	Distance		Utiliztaion	
Scenario	**1-3 Workforce**	**2-6 Workforce**	**1-3 Workforce**	**2-6 Workforce**
1	35,586.09	35,613.80	24.46%	12.28%
2	32,188.07	32,270.04	23.63%	11.87%
3	**31,782.92**	**31,953.76**	**23.50%**	**11.84%**

Table 4.17. Daily Average Total Distance in meters and Utilization

Concluding on the results in Table [4.17], it can be seen that Scenario 3 provides the largest average reduction in distance and utilization when averaging all scenarios. By workforce variation from 1-3 and 2-6, the distances were reduced by 10.69% and 10.28%, and reduced utilization by ≈ 4% for both configurations of the workforce. The reduced utilization can be interpreted as resources being freed due to efficiency gains.

To quantify the reduction in time the current workforce of six has been used.

- U_A = Average utilization As-Is
- U_S = Average utilization To-Be
- T_a = Time available
- W = Workdays in a year

$$\text{Time Saving} = (U_\text{S} - U_\text{A}) \cdot T_\text{a} \cdot W$$
$$\Updownarrow$$
$$((0.1228 - 0.1184) \cdot 76) \cdot 252 = 84.26 \text{ Hours Per Year}$$

A reduction of 84 hours per year is not as much as expected in Section [4.1]. This value is dependent on the model configuration. Thus the actual operation speed might differ. Furthermore, half of the working time is spent on loading and unloading, which is values that are kept static. Thus these values cannot be reduced.

4.2.8 8. Final recommendations

As presented in Section [4.2.7], two new scenarios were analyzed in comparison to the As-Is model. The results showed that distance and time could be reduced in both scenarios by allocating the dock doors based on the demand groups. Based on the results it can be seen that the total reduction in distance and time is achieved by relocating the DK91 PS zone and allocating dock doors based on the demand. Thus it would be recommended to apply both changes as tested in scenario 3.

However, it would be recommended to implement a variable workforce as it was found that the half workforce could handle demand in some demand periods. A further description of how this should be implemented can be found in Chapter [6].

It should be noted, that it was determined in Section [4.1], that the demand is fluctuating from day to day, which implies a fixed dock door allocation is not sufficient; rather a dynamic dock door allocation, where the allocation is reconfigured to match the demand of the given day.
Thus, to transfer the obtained results to real-world operations, knowledge of demand data is required on a daily basis. As mentioned in Section [2.2.1], data quality and availability are not mature and incapable of such a requirement, essentially a live-data information flow is required. To have information about inbound daily demand, would require PNL to have information from upstream in the supply chain, which they currently do not have access to. This will be further elaborated upon in Section [4.2.9]

4.2.9 Supply Chain Visibility

Supply chain visibility (SCV) relates to the 'visibility' or transparency of information flow in a supply chain network. SCV is characterized by information accessibility, accuracy, timeliness, completeness, and usage to enhance operational and strategic activities. [Kalaiarasan et al., 2022]

In general, SCV has become an intrinsic part of supply chain performance within the past couple of decades. Although the term itself has become an integral part of supply chain management, many organizations are challenged by their lack of supply chain visibility. This is either due to lacking technological advancements or managerial insight. As a result of this, managerial deviations are not easily accommodated, as decisions are made reactive rather than proactive. This ultimately leads to poor decision-making that in turn results in lost sales, inefficient use of resources, or other competitive advantages. [Kalaiarasan et al., 2022]

Through Chapter [4], real observations and data were used to do manual calculations as well as simulations, in order to investigate whether or not the current dock door setup at PNL's DC in Aalborg could be changed to decrease complexity, and thereby time spent and distances traversed. After conducting the analysis, improvements were achieved. As already mentioned in Chapter [2] the data quality at PNL is poor, and this especially became apparent when analyzing the KP2 and KP4 scans from the gathered data. From here, it became apparent that 31% of the KP2 scans were missing from the totality.
This is the scan of the cargo as it arrives at any given DC, meaning that 30% of all cargo is an unknown quantity for the next step in the Supply Chain.

Apart from this, the analysis has been based on historical data, which is fine for proof of concept, but realistically, live data is the next step for PNL if optimizing their DC flow. Despite PNL doing an effort to become more data-driven, this is something they are yet to fully achieve. To mention a few things they are lacking, apart from the scans, which in itself is an inhibition - data points such as; the size of cargo, precise arrival of the truck, and a database that only updates once every 30 minutes. So, the lack of these data points means that PNL in general is far from achieving real-time data, meaning that a gap is present between the findings and what is actually possible.

This sign of poor data quality inhibiting planning decisions can be directly linked with lacking SCV as demand information from actors upstream in the supply chain is not released. Ideally, real-time data as a result of increased SCV can result in many competitive advantages, not only for PNL and PN as an organization but lead to improvements for all relevant actors in the supply chain. However, this is a challenging requirement based on third parties and supply-chain actors and increased collaboration in the supply chain, such as electronic data interchange (EDI), to improve information flow. [Premkumar et al., 1994]. To gain information about daily future demand, an implementation of demand forecasting has been selected to mitigate the lack of supply chain visibility.

Thus, the following section will emphasize the implementation of demand forecasting, and how it can help to improve supply chain visibility.

Analysis Part II 5

5.1 Forecasting

In order to try and close the gap between the proposed solution, supply chain visibility, and the lack of live data, forecasting will be implemented. Forecasting is a method that is often used to predict uncertainty and improve planning and scheduling. In general, it helps people and organizations to plan for the future and rationalize decisions. [Armstrong, 2002] When forecasting it is imperative to choose the right model, as the different methods can react differently depending on the nature of the data.

5.1.1 Demand Classification for Forecasting

It is established not all forecasting models can provide applicable demand predictions, as the data structure and the nature of the demand can exhibit very complex patterns or patterns some models are not designed to accommodate. Intermittent demand in particular, which is very prominent in real-world data is difficult to predict and fit an applicable model for [Hyndman and Koehler, 2006; Lei et al., 2017]. Throughout the years, several models have been developed with their own advantages as well as disadvantages. Although examining data before selecting a model is preferable, this can be very resource intensive and intangible by visually inspecting multiple time series.

A method introduced by Croston [1972], referred to as *Demand Classification* can be applied to split demand into groups. This idea has been further developed over the years, including Syntetos and Boylan [2001]; splitting time series demand into four groups to determine demand regularity, and thereby forecastability. [Rožanec et al., 2022]. Demand is characterized as either *regular* or *irregular*, with subgroups to ascertain demand type:

- Regular Demand
    - Smooth: Regular demand with low quantity
    - Erratic: Regular demand with high quantity
- Irregular Demand
    - Lumpy: Irregular demand with high quantity

- Intermittent: Irregular demand with low quantity

[Rožanec et al., 2022]

These four subgroups are defined by fixed intervals between two calculated values, *Average Demand Interval* (ADI) and the *Coefficient of Variation* (CV), which corresponds to four quadrants. The quadrants are formed using threshold values defined by [Syntetos and Boylan, 2001; Croston, 1972], where the cut-offs are ADI = 1.32 and $\mathrm{CV}^2 = 0.49$. These cut-offs can be graphically represented as four quadrants, equating each of the four demand types. How to calculate ADI and CV^2 can be seen in Eqs. [5.1] and [5.2].

$$\mathrm{ADI} = \frac{\text{Total Number of Periods}}{\text{Number of Demand Buckets}} \tag{5.1}$$

$$\mathrm{CV}^2 = \left(\frac{\text{Standard Deviation of a Population}}{\text{Average Value of a Population}}\right)^2 \tag{5.2}$$

$$\text{Smooth} = \mathrm{ADI} < 1.32;\ \mathrm{CV}^2 < 0.49 \qquad \text{Erratic} = \mathrm{ADI} < 1.32;\ \mathrm{CV}^2 \geq 0.49$$

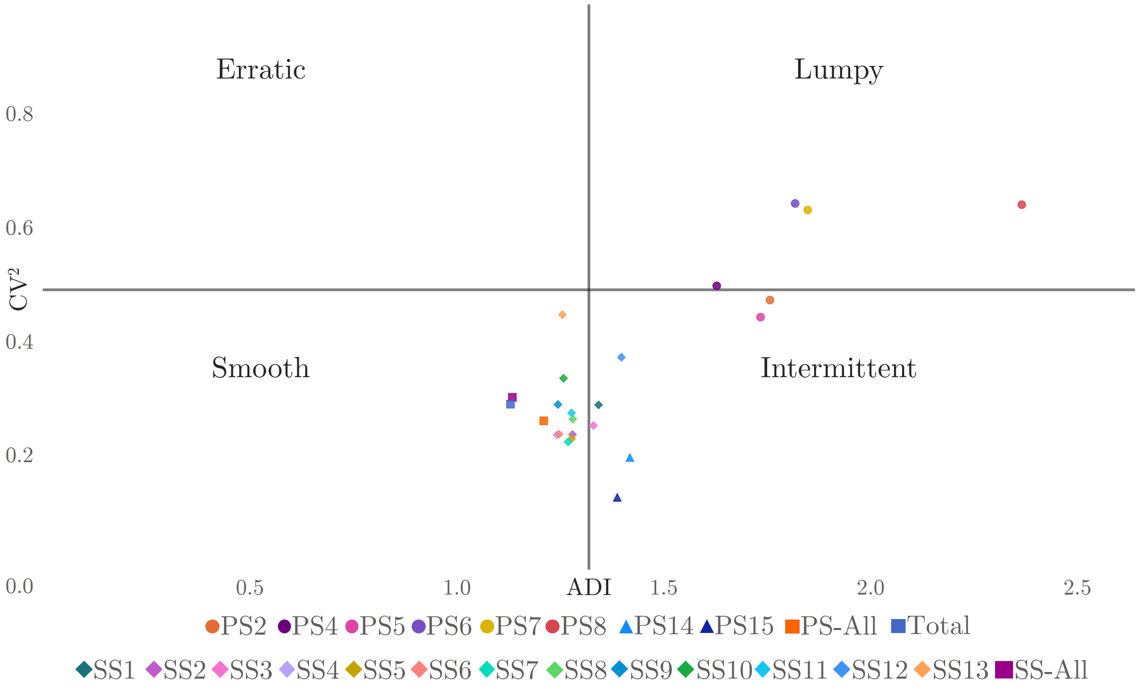
$$\text{Lumpy} = \mathrm{ADI} \geq 1.32;\ \mathrm{CV}^2 \geq 0.49 \qquad \text{Intermittent} = \mathrm{ADI} \geq 1.32;\ \mathrm{CV}^2 < 0.49$$

Forecasting lumpy and intermittent demand that is highly volatile can be challenging as demand is unstable with multiple zero-demand. To mitigate this, forecasters can perform temporal aggregation to reduce volatility, which simply lowers demand frequency by aggregating a time series observation period from hourly to daily, daily to weekly, etc. A disadvantage is loss of information, by altering patterns such as trend and seasonality through aggregation. However, by disaggregating data you similarly risk dealing with intermittent data with low forecastability. [Lei et al., 2017; Dyckhoff et al., 1994]
In order to account for this, lumpy and intermittent demand often requires advanced models. For this specific purpose, Neural Networks, Hybrid-, and other machine learning models are often applied. Erratic and smooth demand being regular does not require advanced models to obtain applicable results. For regular demand, simple regression models like Simple Moving Average or Exponential Smoothing are often sufficient. [Rožanec et al., 2022]

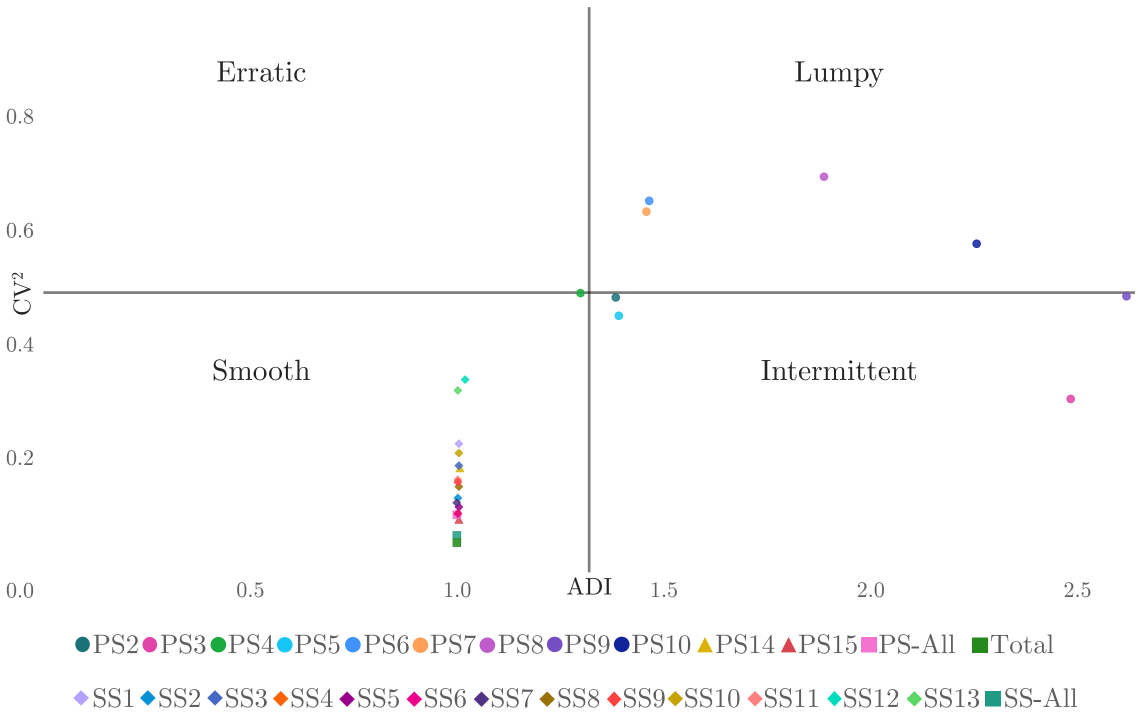
Thus, by applying demand classification to a given demand, it is possible to determine demand regularity and thereby determine forecastability by defining demand as regular or irregular. However, this is not necessarily sufficient, as some data patterns can better be captured by one model, depending on the input data: E.g. one company might have more complex patterns in their intermittent daily demand with multiple seasonality; as opposed to a different company with the same frequency and irregularity but with simple trend

and seasonality patterns. In other words, a single forecasting model that is accurate for one use case, might be insufficient in another - as how 'good' a model is, is dependent on the input data.
Therefore, in this thesis, multiple forecasts methods will be selected, to determine if a particular model, with the given data, is better suited for a given demand type.

Firstly, the provided data will be investigated and demand classification will be performed. Due to the nature of the data, it is expected to have several zero-values, as PNL does not operate on weekends and holidays, meaning a demand of zero for these instances [Assad Mohammed, 2023]. Furthermore, as PS and SS are disaggregated in groups (as visualized in Fig. [4.9]), the aggregation level itself could contribute to zero demand. To accommodate this, demand classification will be applied to unaltered demand and demand where weekends and holidays are removed. Also, in terms of the aggregation level, different levels of demand aggregation will be tested, as it is expected to smooth the demand. Hence, demand is aggregated into PS, SS, and the total demand, dubbed *PS-All*, *SS-All*, and simply *Total* on the graphs in Fig. [5.1]



(a) Demand Classification of Demand Groups (All Days)



(b) Demand Classification of Demand Groups (Weekends and Holidays Removed)

Figure 5.1. Demand Classification Graphs of Forecastability (2021-07-30 - 2023-03-01)

As seen in Fig. [5.1] the demand type for the demand groups is either intermittent, lumpy, or smooth. Some demand groups break the scales of this graph but are still visually represented by Fig. [C.1] in Appendix [C]. It is seen how the data is changing from lumpy/intermittent to smooth as it is aggregated, see Fig. [5.1a]. Furthermore, by removing special days (weekends and holidays), the majority of the demand groups become smooth, as can be seen in Fig. [5.1b].

Type	Smooth	Lumpy	Intermittent	Regular	Irregular
AD	11	7	13	11	20
NS	19	5	7	19	12
PS-AD	0	7	9	0	16
SS-AD	10	0	4	10	4
PS-NS	4	5	7	4	12
SS-NS	14	0	0	14	0

Table 5.1. Demand Classification of Demand Groups: Count of Demand Type

As seen in Table [5.1], All days (AD) have a majority of demand being intermittent and irregular. But by having no special days, i.e. no weekend days or holidays (NS), the demand type is essentially flipped from regular to irregular. It can also be seen, that PS exhibits primarily irregular demand and the opposite can be said of SS. Furthermore, as PS's demand type prevails even after removing special days, it can be deduced that zero demand occurs regularly for the PS demand groups, regardless of the embedded 'natural' zeros due to PNL's operations.

Hence, it is imperative to decide the granularity of the data when forecasting, as information can be lost if the total is used to represent each aggregation level. The difficulty of forecasting increases as the data becomes more granular, due to sparsity, lack of patterns, and uncertainty. [Clark and Avery, 2010] Due to these considerations, the data will be forecasted based on each individual dock door, to better exhibit the nature of the operations on a daily basis. In other words, the hierarchy of the data is considered, instead of forecasting the total demand for the DC.

Moreover, the use of both R and Python to conduct the forecasts also helped to ensure integrity and ease of use, as many of the forecasts exist as packages within the respective programming languages. This helped to achieve faster and statistically proven results. The source code can be found in Appendix [E.4].

5.1.2 Selection of Forecasting Models

As the nature of the demand was investigated in Section [5.1.1], the next step is to find appropriate forecasting techniques that accommodate the behavior of the data, but also consider forecasting maturity of the company. [Armstrong, 2002] has developed six different criteria to support forecast model selection, see Table [5.2], where 5 of them have been summarized, and the last one is presented after the table.

Method	Strength	Weakness
Convenience	Minimum effort put into finding the model for forecasting (ease of use)	Can cause serious disruptions in environments with large change
Market Popularity	Based on shared experience (what worked for others might work for me)	Blindly choosing, not considering the use case, as details can not be interpreted in surveys
Structured Judgement	Well thought out as evaluated criteria are compared to different models	Decisions are made based on expert-statements, which can introduce bias
Statistical Criteria	Accurate in pinpointing exact methods, based on statistical evidence	Presenting absolute values, meaning that details about the model might be missed
Relative Track Records	Mitigates the need to generalize from other research	Assuming that historical results can be generalized to the future

Table 5.2. Criteria for selecting a forecasting method

The last method to consider is written outside of the table, as it contains a more broad consideration of the selection phase, hence the term *General Principles* is introduced.

- **Structured**; Use structured methods as they are easier to replicate and communicate.
- **Simple**; Stick to a simple model unless complexity is necessary.
- **Quantitative**; If the data is sufficient, studies indicate that quantitative methods should be used instead of judgemental.

The first criterion that has been chosen in order to select the most appropriate forecasting method is the statistical criteria. This method is frequently used, as it helps to determine the accuracy of the different forecasts. The performance measures used will be introduced after the forecasts have been presented in Section [5.1.3.3].
Convenience has been used to assess the time and resource aspect, to make sure that the forecasting methods fit. Furthermore, together with Market Popularity it has been used to select methods that are well-known amongst practitioners, and thereby the consensus of use within different areas is much broader compared to the relatively new and untested methods. This has been further supported by the General Principles. Concurrently, General Principles suggested the use of quantitative models. Structured judgment and Track Records have both been discarded, as they required prerequisites that were unobtainable such as previous forecasts and assessments from relevant stakeholders.

5.1.3 Selected Models

It is important to note, that considering the data consists of univariate time series models, the selected models should accommodate such a data. Furthermore, common time series

features, such as seasonality and trend, should also be features such models are capable of capturing.
After applying the selection heuristic, the models that have been selected for forecasting are the following; Auto-regressive Integrated Moving-average (ARIMA), Error Trend Seasonal (ETS), Neural-Network auto-regression (NNAR), and Prophet. A short explanation of their respective contribution is provided.

5.1.3.1 Statistical Models

The statiscal models ETS and ARIMA have both been selected as they are among the most commonly applied forecasting models. Apart from being complementary models, each model focuses on different areas within time series analysis. ETS is based on a description of the trend and seasonality of the data, whereas ARIMA is focused on the autocorrelations in the data (i.e. the similarity between demand observations as a function of the time lag, or shift, between them.) [Hyndman and Athanasopoulos, 2023]. ARIMA and ETS can be considered *traditional* methods.
They are simple and easy to interpret, and have good

autoARIMA
ARIMA is a generalization of the auto-regressive moving average (ARMA(p,q) model, where *integrated* stems from the differencing term (d) introduced in an ARIMA(p,d,q) model. Thus, the ARIMA model consists of the orders:

- AR(p): In an autoregression model, the variable is forecasted by applying a linear combination of past values of the variable. Autoregression indicates that it is a regression of the variable against itself. [Hyndman and Athanasopoulos, 2023]
- I(d): The differencing term relates to the stationarity of a time series. A time series that is stationary, is one whose statistical properties do not depend on the time at which the series is observed. Therefore, time series with seasonality or trends are not stationary, as seasonality and trend affect the values of the time series at different times. The goal of differencing is to help stabilize the mean of a time series by removing changes in the level of a time series, thereby eliminating (or reducing) trend and seasonality. [Hyndman and Athanasopoulos, 2023; Kwiatkowski et al., 1992]
- MA(q): Rather than using past values of the forecast variable in a regression, a moving average model uses past forecast errors in a regression-like model. It should

> not be confused with the moving average *smoothing* approach. A moving average model is used for forecasting future values, while moving average *smoothing* is used for estimating the trend cycle of past values. [Hyndman and Athanasopoulos, 2023]

Thus, a model with p, d and q all equal to an order of 1, would be considered an ARIMA(1,1,1) model.
Determining the given orders of ARIMA is conducted with auto-correlation functions such as ACF and PACF plots, where the order of differencing can be determined by whether it is necessary to apply differencing to the time series if it is non-stationary. Differencing is only one way to make a non-stationary time series stationary. It is achieved by computing the differences between consecutive observations.
To further extend the ARIMA model to model a wide range of seasonality, a seasonal ARIMA (SARIMA) model is formed by including additional seasonal terms to the presented ARIMA model, noted as ARIMA(p,d,q)(P,D,Q)m. The extended parts (P,D,Q), are the seasonal parts and are similar to the non-seasonal model, but include back-shifting of the seasonal period (m), where the seasonal terms are simply multiplied by the non-seasonal terms. Seasonal terms are similarly determined by reviewing the time series, and conducting ACF and PACF plots by focusing on the seasonal lags.
Seasonal lag, noted by m, is usually determined by the time period or type of the time series, e.g. m=4 is quarterly data, m=12 is yearly, m=7 is daily and so on. [Hyndman and Athanasopoulos, 2023]

Due to the complexity of determining the terms of an ARIMA or even SARIMA model, and repeating this for multiple time series, an automated approach has been developed, where the (S)ARIMA order of terms is estimated algorithmically. This approach is referred to as *autoARIMA* and uses several tools to determine test stationarity, and seasonality and find the best model within a range of orders. As it is not possible to graphically (manually) determine both p/P and q/Q terms at the same, autoARIMA can explore more complex models than what is possible manually and can be more useful for the purpose of complex modeling. [Hyndman and Khandakar, 2008; Hyndman and Athanasopoulos, 2023]

Error Trend Seasonal State Space Model: ETS

Exponential smoothing can be viewed as a weighted average, where instead of applying equal weights to all observation (reffered to as a naivë model), the weights decrease *exponentially* as observations come from further in the past. To determine the weights the *smoothing* parameter α is applied. The closer α is to 0, the more importance is given to past values; and the closer α is to 1 more weight is given to the more recent values. For any α value between 0 and 1, the attached weights to the observations decrease exponentially as we go back in time, thus the name *exponential smoothing.* An example of the simplest exponentially smoothing model, called the Simple Exponential Smoothing (SES) model, is given in Table [5.3].

$\alpha = 0.2$

Period	Value
yT	0.2000
$yT-1$	0.1600
$yT-2$	0.1280
$yT-3$	0.1024
$yT-4$	0.0819
$yT-5$	0.065

Table 5.3. Example of a Smoothing Parameter $\alpha = 0.2$ [Hyndman and Athanasopoulos, 2023]

However, SES best captures data that is without trend or seasonality. To handle more complex patterns several models have been introduced over the years, where trend and seasonality can be classified as either additive or multiplicative - however multiplicative trend is often disregarded due to poor performance. [Hyndman and Athanasopoulos, 2023] An example of exponential smoothing methods is given in Table [5.4]

Trend Component	**Seasonal Component**		
	N	A	M
	(None)	(Additive)	(Multiplicative)
N (None)	(N,N)	(N,A)	(N,M)
A (Additive)	(A,N)	(A,A)	(A,M)
A_d (Additive damped)	(A_d,N)	(A_d,A)	(A_d,M)

Short hand	**Method**
(N,N)	Simple exponential smoothing
(A,N)	Holt's linear method
(A_d,N)	Additive damped trend method
(A,A)	Additive Holt-Winters' method
(A,M)	Multiplicative Holt-Winters' method
(A_d,M)	Holt-Winters' damped method

Table 5.4. Different Trend and Seasonality components for Exponential Smoothing [Hyndman and Athanasopoulos, 2023]

Additive is preferred when variations are roughly constant through the series, whereas multiplicative is preferred when the variations are changing proportional to level of the series. For example with a seasonal component: in the additive method, the seasonal component is given as an absolute value scaled by the observed series, and the series is seasonally adjusted by subtracting the seasonal component. For the multiplicative method, the seasonal component is given as a percentage, and the series is seasonally adjusted by dividing by the seasonal component. [Hyndman and Athanasopoulos, 2023]

Since exponential smoothing has many variations, a method to automatically select the 'best' variation is introduced, the innovations state space model, ETS. It is an extension of the the all the possible models of exponential smoothing, in which some were presented in Table [5.4], and is simply denoted by ETS(*Error*, *Trend*, *Seasonal*). For each method, two models exist, one with additive- and one with multiplicative errors. Thereby, the extension of *Error* is applied to distinguish between between prediction intervals. [Hyndman and Athanasopoulos, 2023]

If multiplicative trends are also considered, ETS is capable of 30 variations of exponential smoothing models and automatically select the 'best' model by the given time series input.

5.1.3.2 Machine Learning Models

As described in Section [5.1.1] the demand for the different dock doors was primarily lumpy or intermittent. To account for the nature of the data it has been decided to introduce more advanced forecasting models such as NNAR, Prophet and LightGBM.
They are capable of handling multiple seasonality and non-linear relationships between variables. Thus, such models have proven to excel with complex temporal patterns in data that are challenging to predict. Due to the *black box* and complex nature of these algorithms, a disadvantage of machine learning models is interpretability. [Hyndman and Athanasopoulos, 2023]
The main focus when introducing these models is not explaining the models in depth, but rather their strengths; this is especially true for Prophet and LightGBM.

NNAR

Neural networks are meant to imitate the human brain in the way information is processed. It handles non-linear relationships between the response variable and its predictors. The data is processed through *layers* which are referred to as neurons and or predictors (bottom layer). The other part is the forecast and or output (top layer).
As for time series, the lagged values can be used as inputs to the neural network. This is also called neural network auto-regression. An NNAR model is typically denoted as NNAR(p,k) where p indicates the amount of lagged periods and k is the number of nodes in the hidden layer of the network. The hidden layer, is essentially used to introduce non-

linearity into the model, where $k = 0$ would be equal to linear regression.
For instance, a NNAR(4,2) contains the four last observations of the time series that is used to forecast the value of y_t. The hidden layer consists of two notes. An NNAR(p,0) is equivalent to an ARIMA(p,0,0), without considering the parameters that handle the stationarity of a time series. Furthermore, the model can be expanded to an NNAR(p,P,k)m model that accounts for seasonality. [Hyndman and Athanasopoulos, 2023]

Prophet

The Prophet model is procedure for time series forecasting based on a generalized additive model, created by Facebook. Non-linear trends are fit with yearly, weekly, and daily seasonality. Performance is best when seasonal effects are strong, and several seasons of historical data is given. Prophet is considered a non-linear regression model and consists of the four following parameters; $g(t)$ describes a piecewise linear trend, $s(t)$ accounts for the seasonal patterns, $h(t)$ captures the holiday effects, and the last parameter is ϵ_t for capturing white noise errors. [Taylor and Letham, 2018]
Prophet is flexible, as it can easily accommodate multiple patterns in input data and is very fast, which simplifies forecasting at scale and complexity of applying an advanced forecasting model. [Taylor and Letham, 2018].

LightGBM

Another machine learning approach based on the ensemble model Gradient Boosting Decision Tree, named LightGBM, short for *Light Gradient Boosting Machine*, has been selected. As the name implies, it is based of a successful XGBoost algorithm, where the goal is faster performance for comparable accuracy. This method is suited for high feature dimensional data and large data sizes, and is also capable of classification. [GuolinKe et al., 2017] Since this thesis specifically focuses on univariate time series, a variation of LightGBM focusing on this has been applied, dubbed *LazyProphet* by its author [Blume, 2023].

5.1.3.3 Performance measures

According to [Hyndman and Athanasopoulos, 2023] there are two focal steps to evaluating the performance of a forecast, or rather, the accuracy of the forecast. A forecast can not be evaluated based on how large the true forecasting errors are, as it is not a valid indication. The only way a forecast can be tested accuracy-wise is by seeing how the model predicts

unseen data. In other words, the entire data set should not be used when fitting the model. Hence, a train-test-split has been performed on the data before conducting the different forecasts. The data is typically partitioned 80/20, where 80% is the training data and 20% is the test data. The test set should be at least as long as the forecasting horizon. [Hyndman and Athanasopoulos, 2023]
The other part of ensuring proper testing of forecasting accuracy is to introduce performance measures. Most forecasts contain errors, which is the deviation between the forecasted values and the actual values. These errors can be summed in different ways to test the accuracy. The different measures are split into three different groups; Scale Dependent, Percentage Errors, and Scaled Errors. [Hyndman and Athanasopoulos, 2023]

Specifically, the selected forecasting accuracy performance measure selected is MASE, a Scaled Error, which has desirable properties such as handling intermittent data well, unlike percentage errors such as MAPE. Furthermore, MASE has the ability to compare forecast accuracy between models as it is not scale-dependent. This is achieved by scaling the mean absolute errors (MAE) of the predicted values by the MAE of a 1-step random walk. Thereby it is a scaled version of the MAE. [Hyndman and Koehler, 2006]
Furthermore, MASE penalizes positive and negative errors equally, thus rendering it suitable for comparison between not only forecasts models of a time series, but also between time series. [Hyndman and Koehler, 2006]
Interpretability of MASE is simple, as it is based on the in-sample random walk (also called a *naivë forecast*), which means that a MASE > 1 can be interpreted as the forecast, is on average, less accurate than a naivë in-sample forecast. Thus, MASE $= 0.5$ means the model has doubled prediction accuracy compared to the naivë model [Hyndman and Koehler, 2006]

5.1.4 Forecasting Results

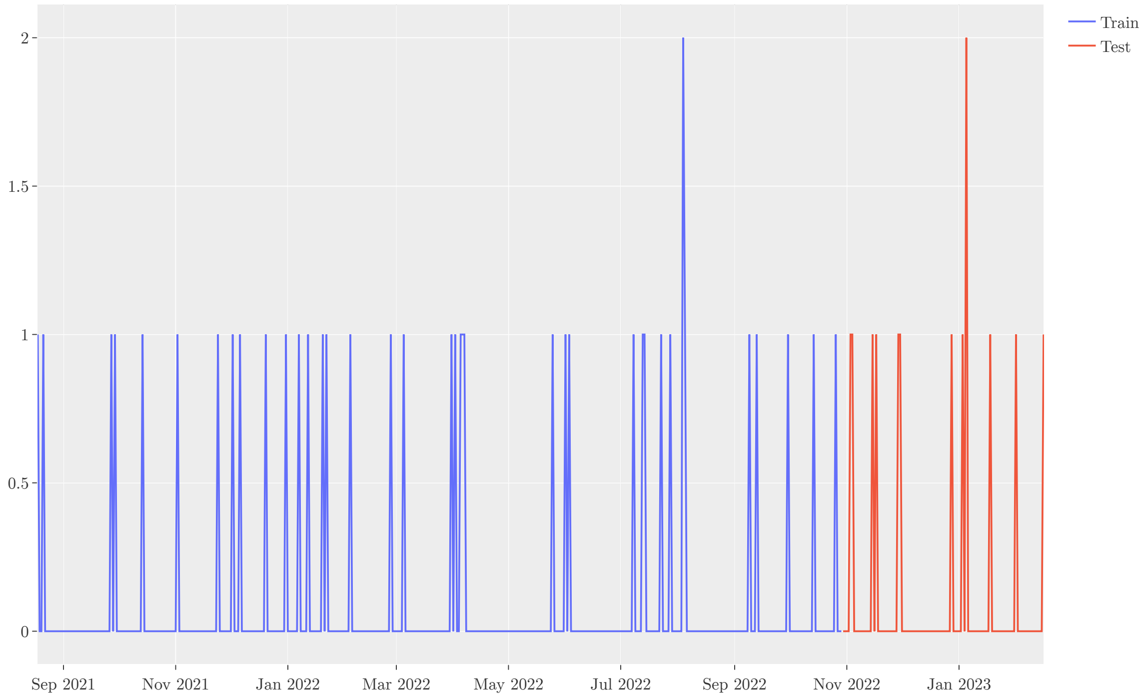
After introducing methods and selecting a performance measure of accuracy, forecasting can be conducted. As mentioned earlier, the complete source code can be found in Appendix [E.4]. The data has been split into each demand group, equalling 28 time series. To gain a better overview, the results will be presented, and some examples will be further examined.

gID	autoARIMA	ETS	NNAR	Prophet	LightGBM
PS1	**0.249**	**0.249**	**0.249**	0.35	*0.36*
PS2	1.089	1.017	**0.867**	1.056	*1.094*
PS3	1	1.032	**0.968**	1.032	*1.21*
PS4	*0.986*	0.883	**0.812**	0.948	0.977
PS5	*1.252*	0.926	**0.904**	1.044	1.141
PS6	0.771	0.938	**0.719**	*1.036*	0.917
PS7	*1.103*	0.922	**0.907**	0.941	0.946
PS8	1.115	1.071	*1.212*	**0.938**	1.027
PS9	**1**	1.156	1.031	*2.406*	*2.406*
PS10	**1**	**1**	**1**	1.054	*1.135*
PS11	**1**	1.115	1.016	**1**	*1.328*
PS12	***1***	***1***	***1***	***1***	***1***
PS13	**0.096**	**0.096**	**0.096**	0.112	*0.131*
PS14	0.286	*0.311*	0.3	0.303	**0.24**
PS15	0.324	0.453	*0.479*	**0.282**	0.326
SS1	*0.367*	0.335	0.368	0.288	**0.255**
SS2	**0.418**	0.431	0.44	0.424	*0.449*
SS3	0.569	0.560	*0.743*	**0.531**	0.575
SS4	**0.408**	0.415	*0.734*	0.421	0.444
SS5	0.512	0.49	*0.542*	0.421	**0.413**
SS6	*0.334*	0.268	0.29	**0.261**	0.308
SS7	*0.433*	**0.378**	0.391	0.399	0.393
SS8	*0.424*	**0.39**	0.392	0.394	0.411
SS9	**0.391**	0.433	0.406	*0.534*	0.433
SS10	**0.352**	*0.422*	0.355	0.398	0.38
SS11	0.329	0.326	*0.367*	**0.322**	0.326
SS12	*0.981*	0.717	**0.638**	0.825	0.737
SS13	*1.005*	0.606	0.586	**0.583**	0.687
gID	**model.autoARIMA**	**model.ETS**	**model.NNAR**	**Demand Type**	
PS1	**ARIMA(1,1,2)(0,0,1)[7]**	**ETS(A,N,A)**	**NNAR(24,1,12)[7]**	Intermittent	
PS2	ARIMA(1,1,3)(0,0,2)[7]	ETS(A,N,A)	**NNAR(24,1,12)[7]**	Intermittent	
PS3	ARIMA(4,1,1)(0,0,1)[7]	ETS(A,N,A)	**NNAR(24,1,12)[7]**	Intermittent	
PS4	*ARIMA(0,1,1)(0,0,2)[7]*	ETS(A,N,A)	**NNAR(22,1,12)[7]**	Lumpy	
PS5	*ARIMA(2,0,1)(2,0,0)[7] w/ mean*	ETS(A,N,A)	**NNAR(19,1,10)[7]**	Intermittent	
PS6	ARIMA(2,0,2) w/ mean	ETS(A,N,A)	**NNAR(23,1,12)[7]**	Lumpy	
PS7	*ARIMA(4,1,1)(1,0,0)[7]*	ETS(A,N,A)	**NNAR(24,1,12)[7]**	Lumpy	
PS8	ARIMA(1,1,3)(0,0,2)[7]	ETS(A,N,A)	*NNAR(25,1,13)[7]*	Lumpy	
PS9	**ARIMA(0,0,0)(1,0,1)[7] w/ mean**	ETS(A,N,A)	NNAR(21,1,11)[7]	Lumpy	
PS10	ARIMA(4,0,0)(2,0,0)[7] w/ mean	ETS(A,Ad,A)	NNAR(24,1,12)[7]	Lumpy	
PS11	**ARIMA(1,1,2)(2,0,0)[7]**	ETS(A,N,A)	NNAR(24,1,12)[7]	Lumpy	
PS12	***ARIMA(0,0,0)(1,0,0)[7] w/ mean***	***ETS(A,N,N)***	***NNAR(22,1,12)[7]***	Intermittent	
PS13	ARIMA(2,0,0)(2,0,0)[7] w/ mean	ETS(A,N,A)	NNAR(21,1,11)[7]	Intermittent	
PS14	ARIMA(4,0,0)(0,1,1)[7] w/ drift	*ETS(A,N,A)*	NNAR(25,1,13)[7]	Intermittent	
PS15	ARIMA(5,0,0)(0,1,1)[7] w/ drift	ETS(A,N,A)	*NNAR(25,1,13)[7]*	Intermittent	
SS1	*ARIMA(2,0,1)(1,1,1)[7]*	ETS(A,N,A)	NNAR(25,1,13)[7]	Intermittent	
SS2	**ARIMA(0,0,0)(0,1,1)[7]**	ETS(A,N,A)	NNAR(24,1,12)[7]	Smooth	
SS3	ARIMA(1,0,1)(2,1,1)[7]	ETS(A,N,A)	*NNAR(24,1,12)[7]*	Intermittent	
SS4	**ARIMA(1,0,0)(2,1,0)[7]**	ETS(A,N,A)	*NNAR(24,1,12)[7]*	Smooth	
SS5	ARIMA(1,0,1)(1,1,2)[7]	ETS(A,N,A)	*NNAR(26,1,14)[7]*	Smooth	
SS6	*ARIMA(0,0,2)(2,1,0)[7]*	ETS(A,N,A)	NNAR(26,1,14)[7]	Smooth	
SS7	*ARIMA(0,0,4)(0,1,1)[7]*	**ETS(A,N,A)**	NNAR(24,1,12)[7]	Smooth	
SS8	*ARIMA(5,0,0)(0,1,1)[7]*	**ETS(A,N,A)**	NNAR(25,1,13)[7]	Smooth	
SS9	**ARIMA(2,0,2)(0,1,2)[7] w/ drift**	ETS(A,N,A)	NNAR(25,1,13)[7]	Smooth	
SS10	**ARIMA(0,0,1)(0,1,1)[7]**	*ETS(A,N,A)*	NNAR(24,1,12)[7]	Smooth	
SS11	ARIMA(0,0,1)(2,1,2)[7]	ETS(A,Ad,A)	*NNAR(21,1,11)[7]*	Smooth	
SS12	*ARIMA(2,0,2)(2,0,0)[7] w/ mean*	ETS(A,N,A)	**NNAR(23,1,12)[7]**	Intermittent	
SS13	*ARIMA(3,1,2)(1,0,0)[7]*	ETS(A,N,A)	NNAR(25,1,13)[7]	Smooth	

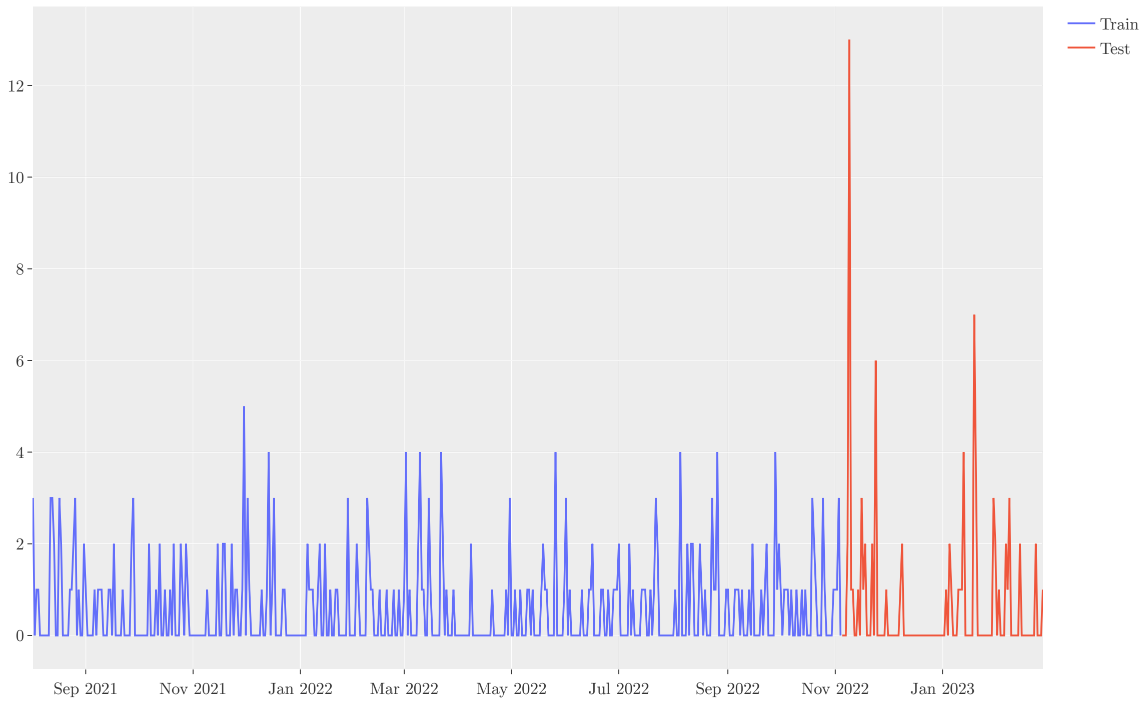
Table 5.5. Forecasting MASE Performance Results for Each Demand Group

In Table [5.5], a table providing all MASE values for each prediction has been provided. MASE values are calculated with a 15-decimal precision, but values in the table have been rounded to three decimal points. The best and worst score for each time series has been marked in bold and italic respectively, and the corresponding model parameters, if any, have also been given. Prophet and LightGBM are the exception, as machine learning models do not necessarily have standardized terms and orders to distinguish between trained models, due to their complexity. A wide variety of models has been selected for autoARIMA, ETS, and NNAR. autoARIMA are primarily seasonal models, with the exception of PS6. For ETS, all models are additive, with the majority being ETS(A,N,A), which are exponential smoothing with additive seasonality and additive errors. All selected NNAR models are seasonal with one lag and between 19-26 lagged periods and 11-14 hidden layers. The 19-26 lagged periods equals approximately a month in days, when weekends and holidays has been adjusted for, which might imply a complex monthly pattern has been determined by the model. Furthermore, the seasonal lag of 1, would imply a weekly seasonality, due to freqyency period, m, is 7 for the time series.

Reviewing the MASE results, several ties between methods can be found, namely in *PS1*, *PS9*, *PS10*, *PS11*, *PS12*, and *PS13*. By inspecting the MASE values, of for example *PS12* as a standout it can be interpreted as none of the models performed better than an in-sample random walk, which is likely due to a suitable model was not found.



(a) PS-12 Demand - Train/Test Split (Intermittent)



(b) PS-10 Demand - Train/Test Split (Lumpy)

Figure 5.2. Train Test splits from Forecasts)

Simply by viewing the graphs in Fig. [5.2], it can be seen that forecastability is low, especially for Fig. [5.2a] which shows PS-12, has been classified as intermittent demand. For Fig. [5.2b], PS-10 Demand is represented and has been classified as lumpy demand. PS-12 demand has many zero-demand buckets and small demand sizes, which is very difficult to forecast, as there seemingly is not an obvious pattern in the demand. PS-10 demand exhibits some patterns and has a relatively steady demand in the train split, however, forecast accuracy will undoubtedly be affected by the demand spike at the start of the test split, which is accompanied by approximately a months gap with no demand, overall not matching well with historical demand. These specific demand groups have been selected to exhibit, that even the most advanced forecasting techniques would probably be unable to model and predict the test horizons - either from a lack of data, low aggregation, or by unexpected demand patterns.

Thus, a well-fit model does not necessarily forecast well, as the predictability of an event depends on several factors, such as: how well the factors that contribute to an event are understood, data availability, and how similar past values are to future values. [Hyndman and Athanasopoulos, 2023]

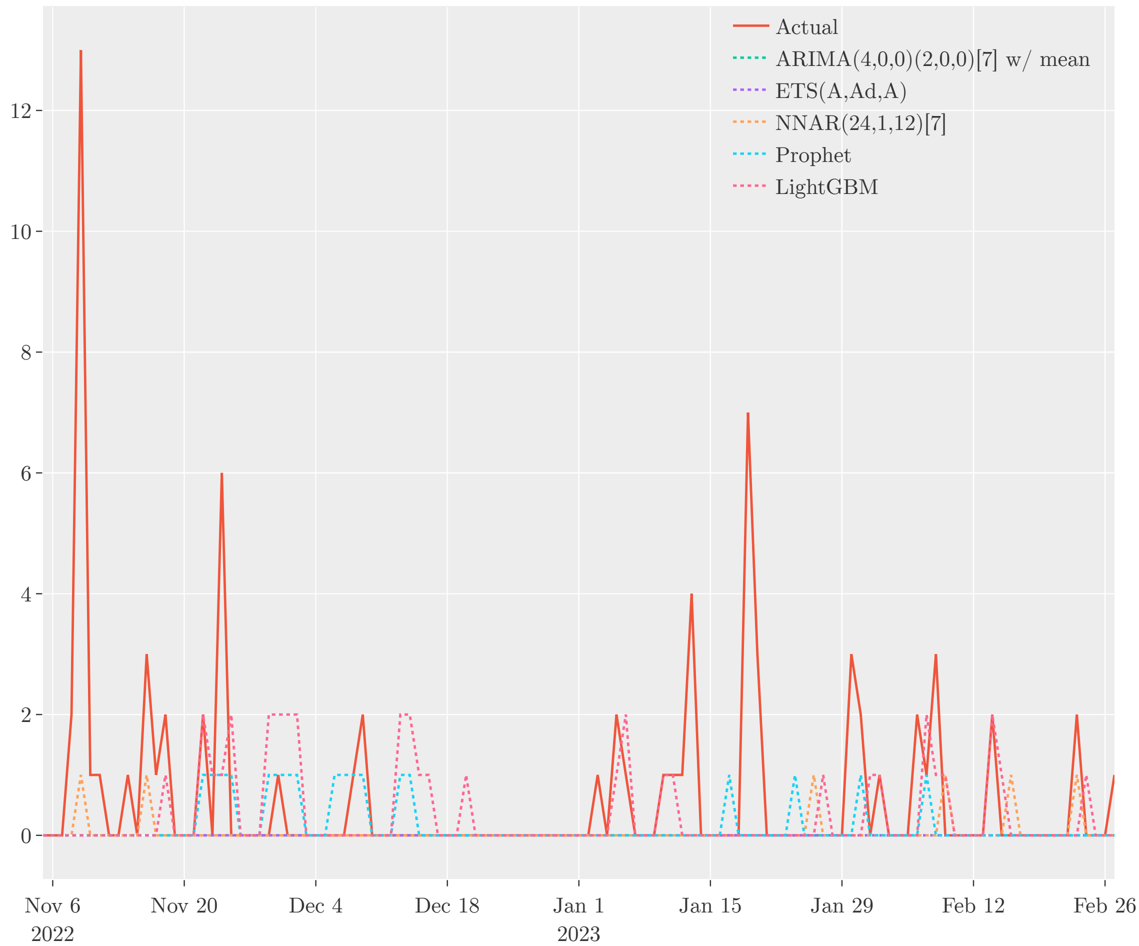
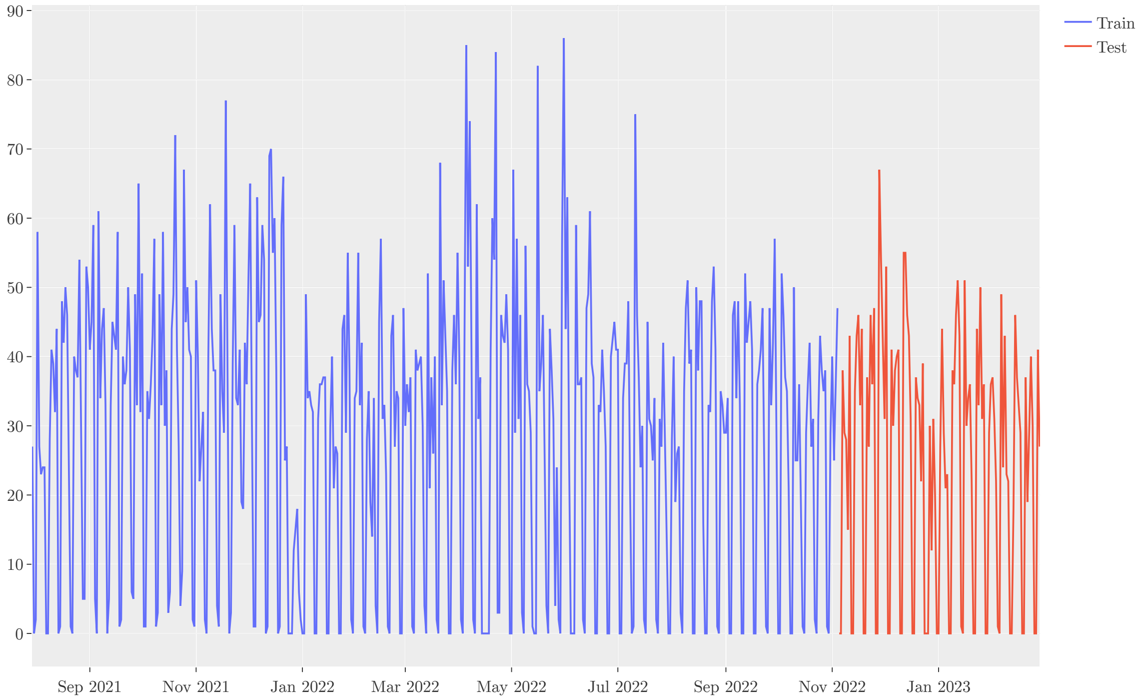


Figure 5.3. PS-10 Demand - Forecast Predictions (Lumpy)

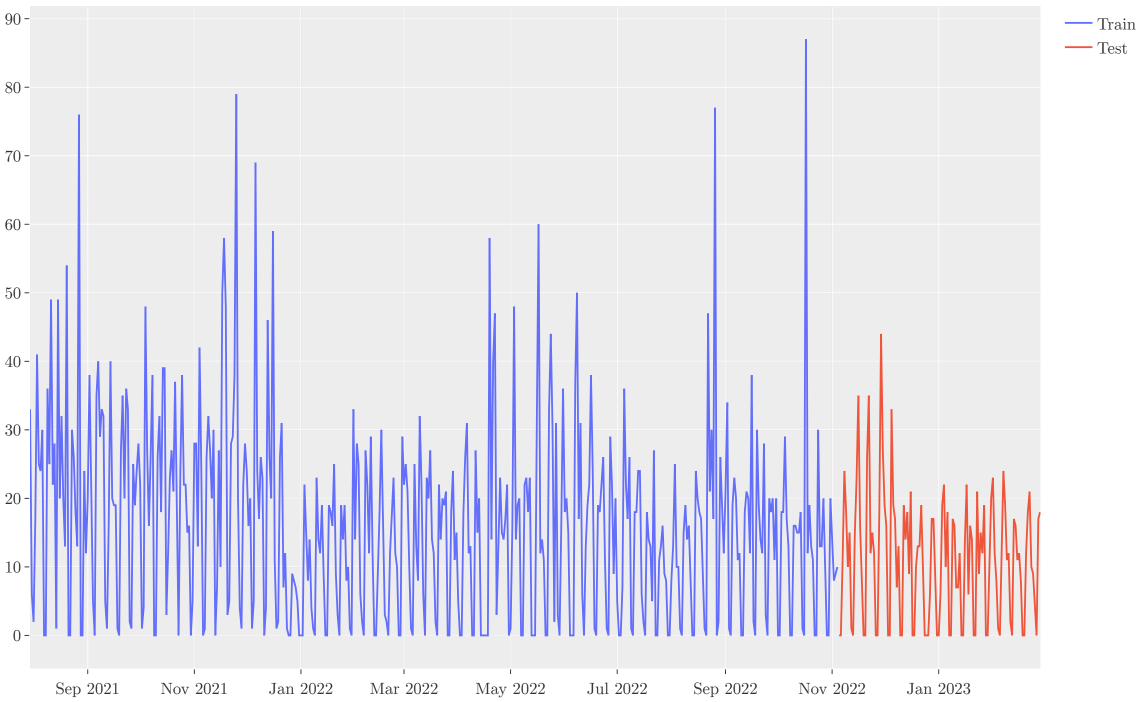
In Fig. [5.3] the predictions of Fig. [5.2b] are illustrated. For this demand group autoARIMA, ETS, and NNAR are tied for 'best accuracy' of 1. This essentially means they are equal to the random walk, which can be attributed to a poor model fit, or the fitted model on the training data has different patterns than the test data; in which both seem to be the case for this time series.

Furthermore, in Table [5.5] *PS13* and *PS1* seemingly are among the best forecasts, but this is misleading. The random walk that dictates the scale of these forecasts is a 1-step forecast, meaning it applies the last observed value as the predicted value. This poses an issue, that is most evident for *PS13*, which has been illustrated by Fig. [D.1] in Appendix [D]. Essentially demand buckets are small, and the random walk has a steady prediction of 3 for the predicted horizon, which is poor as the average demand for the test period is ≈ 0.27. Thereby, the forecast models are 'better' relative to the random walk (as it has fewer errors), but the real-world applicability of the model predictions is still poor for this particular time series, which is reflected by the intermittent nature of the demand.

Some poor examples and expected issues have been outlined, as forecasting irregular demand, as stated throughout, is particularly challenging. To demonstrate how the models handle smooth demand, the 'worst' and 'best' performers in a smooth demand time series will be reviewed.



(a) SS-6 Demand - Train/Test Split (Smooth)



(b) SS-13 Demand - Train/Test Split (Smooth)

Figure 5.4. Train Test splits from Forecasts)

In Fig. [5.4] the best performing smooth demand time series, illustrated in Fig. [5.4a], and the worst performer, seen in Fig. [5.4b], can be found. What is of particular interest, is by comparing the time series, it seems SS6 has a relatively steady mean over time when compared to SS13, the latter exhibiting multiple spikes in the training data. Furthermore, the training and test data is more closely resembled in SS6 compared to SS13; where these factors, as previously established, all affect predictability.

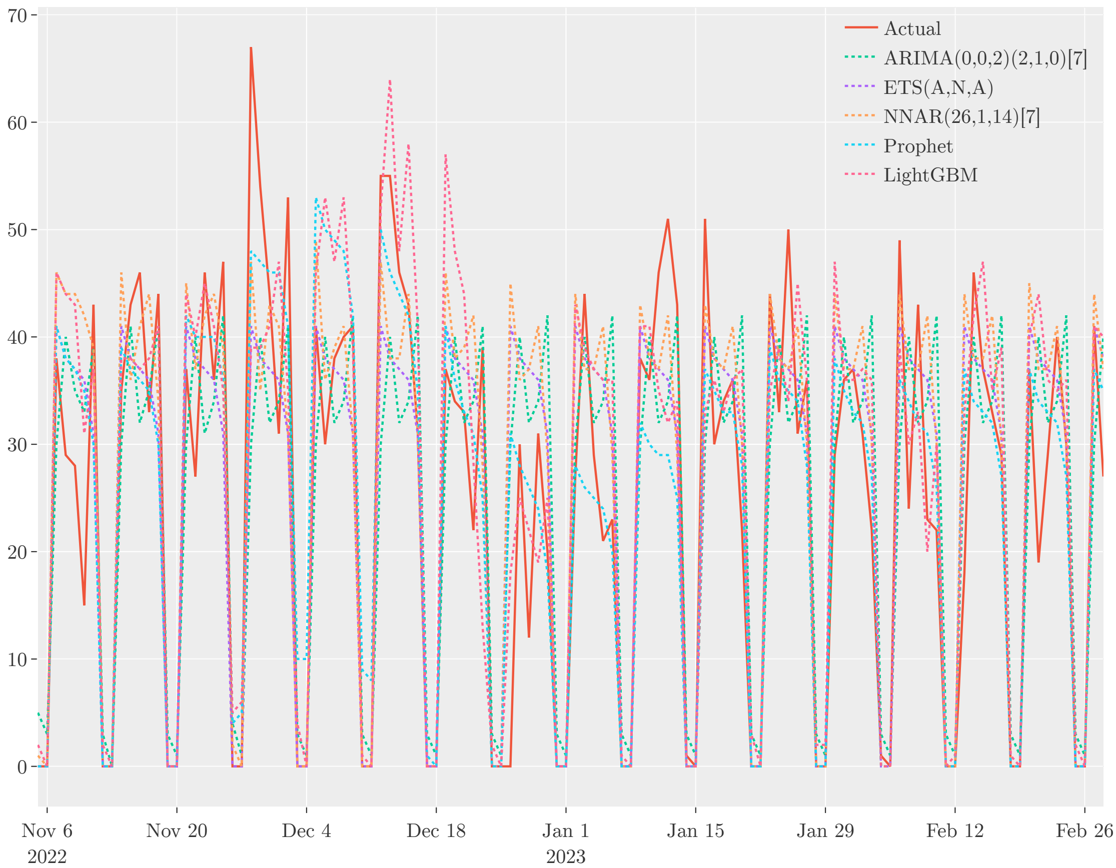


Figure 5.5. SS-6 Demand - Forecast Predictions (Smooth)

The predictions of SS6 have been provided in Fig. [5.5]. It can be seen, that all models have correctly predicted the 5-day work-week, where weekend demand is always zero. All models seem to have similar fits, but Prophet has the comparably the lowest MASE, even though it seems LightGBM has a better fit and captures the nature of irregular demand fluctuations, the spike is off-set; and the more conservative prediction by Prophet has lower error values and thereby better accuracy.

Some of the time series and their predictions have been presented, mostly to present the nature of the data and of the presented accuracy values. But as there are 28 time series and 56 accompanying plots, each and every plot cannot be visually reviewed and commented upon. The individual MASE values have all been presented in Table [5.5], where each time series for each model has been evaluated. To further evaluate performance, it has been chosen to see if demand classification is useful not only to determine forecastability but also a forecast model.

	Best					Worst				
Demand Type	ARIMA	ETS	NNAR	Prophet	LightGBM	ARIMA	ETS	NNAR	Prophet	LightGBM
Smooth	**4**	2	0	3	1	**4**	1	3	1	1
Lumpy	1	0	**3**	1	0	**2**	0	1	1	**2**
Intermittent	0	0	**4**	2	2	3	1	2	0	**4**
Irregular	1	0	**7**	3	2	5	1	3	1	**6**
Total	5	2	**7**	6	3	**9**	2	6	2	7

Table 5.6. Best and Worst Performers per Demand Type

In Table [5.6], an overview of the best and worst performers are summarized. For each demand group, it has been counted which model performed best and worst, excluding the time series that resulted in ties. Since there is no 'Erratic' demand, a complimentary 'Regular' demand type has been omitted from the table.

As seen in the table, it seems NNAR is the strongest model for irregular demand, and autoARIMA is the best for smooth (i.e. regular demand). However, autoARIMA has equal 'best' and 'worst' performances. Therefore, only considering absolutes might not be a viable method of evaluating model performance. A common practice in forecasting is measuring the average accuracy, which is applicable due to the properties of MASE.

Demand Type	autoARIMA	ETS	NNAR	Prophet	LightGBM
Smooth	*0.461*	0.416	0.450	**0.416**	0.424
Lumpy	0.996	1.012	**0.957**	1.189	*1.248*
Intermittent	*0.712*	0.660	**0.652**	0.671	0.694
Irregular	0.829	0.805	**0.777**	0.885	*0.922*
Total	0.693	0.661	**0.656**	0.711	*0.738*

Table 5.7. Average Forecasts Accuracy by Demand Type

As depicted in Table [5.7], NNAR retains the best performing model for irregular demand and overall on average. autoARIMA is now the worst-performing model of smooth demand on average, where Prophet has the lowest errors. It seems despite LightGBM arguably being a more complex algorithm, it has the worst performance. From reviewing plots, there are three possibilities for poor performance, that are applicable to all of the selected models:

- Test and training data are not closely resembled
- The demand did not exhibit recognizable patterns
- The model was over- or underfitted (requiring more attention to parameters)

It can be concluded that NNAR rendered the best results for irregular demand and overall on average, whereas Prophet had the best accuracy on average for smooth demand. Forecastability was determined by demand classification, which was evident in accuracy measures, as irregular demand yielded lower forecast accuracy on average. It could be argued, that one model might still not be sufficient for a given demand group, as the best performer for irregular demand, NNAR, had poor performance or was unable to find a model fit, namely in *PS1* and *PS12*. Predictability was poor for irregular demand, as it is challenging to predict. The most important assumption about forecasting is that future demand and subsequent patterns are associated with historic demand, which might not be true at this aggregation level or the pattern is very complex.

Solution Proposal 6

The following chapter acts as a proposal for PNL, by applying the findings from the selected DK91 DC in Aalborg, and a recommendation for how the identified problem should be implemented in the organization.

6.1 Issues Inhibiting Implementation

In Section [4.1], it was established that by altering dock door allocation to match demand, such that demand was distributed according to the shortest distance to the inbound dock doors; DC efficiency increases. As found in Section [2.1], the highest cost of a DC is the labor cost, which can be split into travel- and handling time.
Thus by reducing these factors, DC efficiency can be increased, and incurred costs can be reduced. As a mathematical model is built on many assumptions, to mitigate some of the flaws and assumptions of a mathematical model, a simulation model was introduced which provided similar results, as travel distance, and time was reduced. From these findings, it is suggested to implement demand-based dock door allocation according to the shortest distance to inbound dock doors.

However, this involves two issues:

- **Issue 1**: Demand is fluctuating from day to day.
- **Issue 2**: Incoming demand quantity on a given day is unknown.

Challenging the current As-Is fixed dock door layout with another fixed dock door layout would not provide the intended results as provided by the simulation scenarios. As established in Sections [2.2.2.1] and [4.1], demand is fluctuating from day to day, and due to a lack of information about incoming demand, the demand on a given day is also unknown. Thus, from the first issue, it is implied that the allocation should be applied dynamically, such that the demand for the given day determines the dock door allocation from day to day.

However, this suggestion is inhibited by the second issue. This inhibition is seen as a

holistic issue, that many organizations face relating to supply chain visibility, as described in Section [4.2.9]. For SCV, actors in the supply chain withhold relevant information that not only benefits an actor such as PNL but would provide an operational advantage to all actors in the supply chain.
This information is most often not held back intentionally, but due to technological limitations or lack of data collection. Technologies such as EDI and API for fast and automatic information sharing are preferred to enable live-data sharing [Sim, 2000; Sumah et al., 2020].
However, such an approach requires investments into systems that are capable and compatible with existing, or even new, systems for all relevant actors in the supply chain. Thus, PNL and the supply chain have a low technological maturity, which inhibits the proposed implementation.

6.2 Mitigating Issues

As mentioned earlier, this is not a unique challenge and can be found in many modern supply chains. Thus to mitigate information that is not available, the common practice is to predict the information. Therefore, forecasting was applied in Section [5.1].

Due to the nature of the data structure, through demand classification, demand groups exhibited irregular demand, which meant forecastability is low. This was reflected in Section [5.1.4], as at the given disaggregation level, predictability was low and would inhibit a dynamic dock door allocation setup.

Thus, the following suggestion for PNL is given:

- Implement a demand-based dock door allocation according to the shortest distance to inbound dock doors.
    - Forecast daily demand to determine highest demand dock doors.
    * Receive daily demand information to determine the highest demand dock doors

In the solution proposal, two variations are suggested, where supply chain collaboration and data maturity is considered. One is for forecasting and the other requires daily demand data from relevant supply chain stakeholders, which is marked by '∗'.

However, to implement the ∗ several requirements are necessary:

- Collaborate with relevant stakeholders and receive demand information
- Data quality
    - Truck arrivals

- Demand Quantity for each truck
- Standardized unit for Demand Quantity (Colli)
- Visual system to reflect dynamic dock door allocation setup

To comment on the stated requirements, it not only pertains to stakeholders but also prerequisites from PNL. Collaboration within the supply chain and increasing SCV have been described, but another issue within PNL is the data quality, as described in Section [2.2.1]. Truck arrivals relate to inbound and outbound demand (line-haul, first-mile, and last-mile), and the lack of visibility within their own system.

As trucks link the supply chain network, it is important to have information about when and where the trucks are in the network, which is often managed manually by a fleet manager. By automatically tracking geographical position in real-time for each truck and driver, could improve route planning and provide transparency to customers upstream and downstream about arrival estimates, re-route as part of traffic, etc.

However, another challenge is the actual demand the trucks are transporting. Due to a lack of transparency, the arriving demand of some days is unknown; it could be a septic tank or a box on a pallet. In other words, what demand a given truck arrives with at a DC, as to what and how much, is unknown. Therefore, demand information about a truck's cargo and a standardized unit of the cargo is a requirement.

Lastly, to implement a dynamic setup, workers would not be able to rely on the routine or fixed labels on the dock doors to distribute inbound demand. Thus, a visual system such as digital monitors above each dock door, or simply implementing a dock door label to their digital scanners is required. The latter option is seen as the better option, as would also increase scan rate, as distribution of incoming demand is dependent on the dock door setup; and this information would be acquired when a scan is performed. Thus, eliminating missing scans and improving data quality. It would also reduce the training time of new and flex workers, as information about cargo placement is provided directly by the digital scanner. As efficiency increases, utilization decreases. Therefore, the added benefits of improved information for planning, work schedules, and workforce could be streamlined according to demand, further reducing costs.

To summarize the given prerequisites, the focus is improving not only SCV, but also internal visibility and strengthening support structures and data quality to ultimately improve planning and flexibility, not only in PNL DCs but throughout PN as an organization and relevant stakeholders in the supply chain.

Finally, the solution proposal for PNL to increase efficiency and flexibility is to implement a dynamic dock door allocation setup and apply forecasts to determine dock door demand. As maturity and SCV are gradually improved and the stated requirements are fulfilled, implementing and utilizing real-time data and information to boost supply chain and organizational advantage, is viewed as the best option that can improve many areas of the operation that is beyond the scope of this thesis.

Discussion & Reflection 7

This section seeks to reflect on and discuss decisions that had an effect on the results. With the purpose of understanding how it impacted the thesis.

7.0.1 Choice of DC

As introduced in Section [2.1.1], PNL is operating multiple DCs nationally. Through Chapter [2], efficiency, demand, and correlation were highlighted in order to choose which DC to do further analysis. It became apparent, that the size of the DCs were associated with efficiency. Interviews with the PNL representative stated that the difference in efficiency was a result of the larger DCs having greater demand. This statement could not be empirically proved through a correlation analysis. Instead, theory from Chapter [2] suggested size as a catalyst to the decreasing efficiency of the larger DCs, which according to [Hackman et al., 2001] is the main factor. From this, the Aalborg DC was chosen for further analysis. However, it might seem counter-intuitive to select Aalborg, as the DC was among one of the best performing DCs efficiency-wise. So, it could be argued that potential improvements could be better at one of the other DCs.
But, through an assessment of size, available time, and interviews, it was decided to delimit the other DCs and focus solely on the DC in Aalborg. By focusing on other DCs, other results, and potentials for improvement might have been discovered, as actual operations at the other DCs have not been observed, meaning other inhibitors than size could affect DC efficiencies. As stated in Chapter [2], there is a relationship between efficiency and the size of a warehouse. Thus, by selecting a larger DC, improvements could have been greater with regard to distance and time reductions.

7.0.2 Data

The following subsection will discuss how the gathered data, as well as the lack of data, has affected the project.

In the Section [2.2.1] the data provided by PNL was described. Through initial data analysis, it was found that $\approx 31\%$ of the KP2 scans were missing. This is significant as the KP2 scans were later used to define the arrival rate for the simulation. Thus the missing data could skew the distribution and thus the arrival rate. Realistically, the exact arrivals of all cargo would have been known, if the scan rate was 100%. In this case, the system would also have been a greater representation of reality. Another issue, was that the exact number of workers on a given day, referred to as the workforce, is unknown. It was determined through interviews that the preliminary flow was conducted by two workers, whereas the secondary flow was conducted by six workers. Although this was stated, the number of workers is not fixed, due to factors such as illness or demand fluctuations affecting the work schedule.

Ideally, the exact number of workers in the DC from day to day should be known, but unfortunately, this is not tracked by PNL. If this was known, it would have enabled modeling of the exact workforce and schedules, and more accurately representing the real-world system, in collaboration with the KP2 scans.

The arrivals of trucks, as well as their carried quantity, were also missing. This meant, that the cargo has to be randomly distributed between some of the dock doors, as the exact dock door for delivery was simply not known. For example, Table [4.6], shows how the demand has been spread equally throughout the dock doors for PS. This problem is not as big for the PS process as it is for the SS process, as the distances from the dock doors to the off-loading zones are the same, but ideally, this should also be a known factor to fully implement the simulation model.

7.0.3 Mathematical Model

As previously known to the authors, the implementation of simulation can help to model reality and thereby account for changes that would be too expensive to implement without considering its actual contributions to the system. Apart from this, the disruptions it can cause if the implementation is based on poor decision-making are also very important.

However, a simulation can be resource intensive as it requires expert knowledge about processes, capabilities, and measurements of the system, which is time-consuming to model. Hence, a mathematical model can be applied as a forerunner to test a system change. The mathematical model showed promising results, with distance reductions up to 20%. Although this seems excellent, there are a lot of parameters that are left unaccounted for when illuminating a complex problem, in which some have been accounted for in Section [4.1.4]. Despite this, the method is believed to give insight into the possibilities of improvement by simply just highlighting the problem. In this way, PNL is introduced into different solutions, each with its own advantages and disadvantages. Due to the simplicity of the method, the mathematical model could have been applied to the other DCs as well. One shortcoming of the mathematical model is its simplicity. As a limited number of parameters are typically applied to the model, a change in one of them might change the outcome of the model completely. An example of this is the speed of the forklifts. The average speed of the forklift was set to 5km/t, as it was believed to represent its average speed within a warehouse environment. As this has been subjectively decided, the speed might as well have been 7km/t or even 10km/t an hour; or an average might not accurately represent the forklifts of the system.

7.1 Simulation

Throughout the simulation of the PNL terminal in Aalborg, a lot of decisions were made with regard to the configuration of the model. These are decisive for the model's performance, and hence the results. This section seeks to reflect and discuss the reasoning and impact of the decision that were made throughout the study.

7.1.1 Arrivals

As the flowcharts and understanding of the model had been achieved in Chapter [2], the simulation section was primarily focused on building the system. The first part was the arrival of goods at the terminal. It was found that PNL had no record of the arrival or quantities of inbound or outbound goods. Therefore it was chosen to utilize the KP2 scans to define the arrival rate. However, as mentioned in Section [2.2.1] the KP2 scans were missing for 31% of the orders. The influence of missing data can be skewed distributions or otherwise misleading patterns. Furthermore, it was found through interviews that the KP2 scans are not necessarily performed at arrival, which further distorts the distribution. Through interviews, it was found that the KP2 scans were sometimes skipped if the cargo is to be transported through multiple DCs. In such a scenario, the scan is not performed before it arrives at the DC closest to the delivery address. Another reason for the poor scan rate is the fact that the importance of the scans is not communicated to the workers [Assad Mohammed, 2023]. Hence there might be a deviance between the actual time of arrival and the scan time.

The Python script that was developed for creating the arrival data, introduced aggregation bias to the data. This was seen as a necessary compromise. This meant that the script could not function on e.g. daily data as there weren't enough data points. It was chosen to calculate the average arrivals per day per hour within the specified time. Furthermore, it was chosen to round up values to the nearest integer. Therefore, near zero values would have the value of 1 which impacts the distribution. As a result, the relation between PP and SS demand was skewed. Ultimately, this impacts the model and the results as the distance traveled varies. It was later found that the Enterprise Dynamics Poisson distribution was capable of using non-integer values. This might have contributed to a more representative distribution.

The arrival of goods to the PS was distributed through a Bernoulli function. This was done to ensure an equal distribution of goods across all inbound dock doors. In practice, this might not be representative, however, as there was no data available it was seen as sufficient. It could be argued that drivers might have a tendency to prioritize parking at the dock door that is closest to the entrance to the site. If this was the case the ideal position of the Queues should be adjusted accordingly.

Lastly, all demand spawns at the first second of every hour, if data had been available on truck size and arrival times. It could be distributed across the hour. Ideally, it would arrive in batches of 8-33 pallets to reflect the different truck loads. However, as the arrival list is dependent on arrivals being in a sequential time order, a random distribution might affect the system negatively.

7.1.2 Warmup Time

In Section [4.2.6] it was found that a warmup time impacted the measurements and thus affected the results. Furthermore, it was found to be sufficient to use a 00:00 to 23:59 cycle. However, results later illustrated that pallets got stuck in the system. It was evaluated that it might be caused by the PS team that starts work at 23:00 and the model ends at 23:59. Thus the PS team only has one hour to handle the DK91 PS demand. To mitigate this issue it would be a requirement to solve the issue of the Warum-time being included in the results or lagging the model to ensure that the shifts are not interrupted. E.g. start with PS at simulation hour 00 and scale the starting time of SS accordingly. As the pallets that are left in the DK91 storage zone are not handled by the SS team, it will have an effect on the results. However, it is difficult to quantify as it might be bound for one of the dock doors within close proximity of one far way. Alternatively, the average value could be used to mitigate this issue.

7.1.3 Advanced Transporters

The advanced transporter atoms were configured based on the top speed of the electric forklifts. This might be misleading as the actual top speed can be reduced by PNL. However, it does reflect the potential of the forklifts. Furthermore, it was determined to use static loading and unloading times of 15 and 5 seconds respectively. These were deemed representative through interviews, however, it could be argued that the nature of scanning an item and identifying the postal code is a stochastic process where the time might vary, depending on the individual worker. Therefore it could be more representative to utilize e.g. a binomial distribution to define the load and unload times.

Furthermore, there is a discrepancy between the logic behind the mathematical model and the simulation model. The demand for one of the PS flows has been placed differently in the two setups. For the mathematical model, it has been placed at dock door 6, whereas it has been placed at dock door 8 in the simulation model. This could be due to lack a of revision, and will ultimately lead to a small discrepancy, but as stated in Section [4.2.5], the process of making a simulation model is iterative, so small deviations between the real world and the model are likely to happen and can be adjusted for in new runs.

It was chosen to use a bidirectional network in the model. This reduces model complexity and further aligns the current operations, as there are no indications of separate transportation lanes. However, it could be argued that two transporters in a bidirectional setup, would have to reduce their speed and further coordinate if they are operating in close proximity. Thus, it would likely add process time to the real-life system.
To mitigate this issue a model could be made with separate lanes to ensure that the Advanced transporters do not interfere with each other.

7.1.4 Dock Door Allocation

In the Section [4.1] it was chosen for the solution to utilize the same dock doors as the ones that are currently used by PNL. However, it was later found through interviews that PNL could utilize the other dock doors if need be. Thus it was determined for the simulation to utilize the dock doors within the closest proximity to the product entry.

Therefore a deviance in the operational flow is present between the mathematical model and the simulation model. This also impacts the comparison as it means that the results are based on different principles. Ideally, a simulation model would be made with the same setup to compare the results one to one. However this was not performed, and thus it cannot be proven how it would have performed.

As seen in Section [4.2.7], the dock door allocation has the greatest impact on the distances. Thus, it could be assumed that utilizing dock doors that are further away, could impact the results negatively.

7.1.5 Solution Scalability

This thesis documents that it is possible to reduce distances and time spent on transportation within the DCs. The test was performed on one of the most efficient and smallest DCs that PNL currently operates. Thus it could be assumed that the potential for improvement might be greater at one of the larger and less efficient DCs. Through interviews, it became apparent that the current processes are similar across all DCs. Thus the framework should be scalable to other departments, although layout and process differences must be accounted for. Furthermore, the data quality suggestions mentioned in Chapter [6] would be beneficial for all DCs as it would allow for better planning of the workforce, routes, and dock door allocation.

7.2 Forecast

In this section, forecasting and its implications in this thesis will be discussed and reflected upon.

7.2.1 Alternative Forecasting Methods

When conducting the forecasting; ETS, ARIMA, NNAR, Prophet, and LightGBM were the forecasts models of choice. ETS and ARIMA were chosen because they were complementary and also frequently used among practitioners. The three other models were chosen as they are able to model complex patterns and non-linear relationships in the data.

In Section [5.1.1] the demand displayed different behaviors both based on aggregation level and group - most of them were either lumpy or intermittent with some being smooth, but very close to intermittent. Common for irregular demand are that the data can be hard to predict, due to zero demand buckets, as well as fluctuations. As it turned out, the more advanced models outperformed both ETS and ARIMA in representing the complex behavior of the data. This leaves the question of whether or not the right amount of forecasting models for the specific purpose were chosen. Nevertheless, some of the models showed MASE values between 0.4-0.6, which means that the model has double the accuracy, compared to a normal random walk.

7.2.2 Forecasting Applicability

Forecasting was introduced as a way to accommodate the lack of supply chain visibility that PNL is experiencing. This is found to be a stopgap solution to a more holistic challenge. However, forecasting is still a viable solution for long-term planning and organizations could also share predictions between them. However, forecast accuracy and predictability are not assured in the short-term and a system capable of handling real-time information streams is preferable. Forecasting can still be an effective tool to bridge the gap between current operations and the lack of visibility, but its use depends on the planning horizon. For example, manufacturing companies would also benefit from live data. But, as the lead time of goods is more likely longer, the day-to-day operations are better accommodated by a forecast and can more easily be changed. This is not the case at PNL, as the idea was to change the dock door allocation on a day-to-day basis. Rather, PNL can benefit from the forecast in such a way, that they can look at past data, forecast it, and concurrently look at the results from both Section [4.1.2] and Section [4.2.7] to further advocate for simple changes, to begin with. With this, it is important to note, that the solutions proposed throughout this thesis, are all based on historical data. This further substantiates the possibilities that can be achieved, just by analyzing the distribution of data from a reactive perspective.

Although, as seen in Table [5.5], the performance of the forecasts were fluctuating, which also infers inaccuracy. Despite this, there could be something worth obtaining, if the forecasted values were used as input data for the coming dock door allocation setup. Instead of applying the fixed As-Is setup, there is a potential gain in implementing the forecasted values instead.

Furthermore, for the purposes of mitigating supply chain visibility, this level of disaggregation increases the average demand interval, as forecasting all 28 demand groups independently, might not be necessary, as the purpose of the daily dock door demand is used for a dynamic dock door allocation strategy. Thereby, forecastability, as well as forecast accuracy, could be increased by aggregating PS and SS dock door demand.

Conclusion 8

The purpose of this chapter is to conclude on the findings of this thesis as a whole and is structured by answering the problem statement and sub-questions stated in Chapter [3].

It was found, that distribution centers constituted 27% of the distribution cost of operations, which is the second highest cost factor. The highest cost factor in a DC is labor cost, which is split into travel- and handling cost, which is linked to DC efficiency. PNL operates in a highly competitive industry with low-profit margins and declining financial results. To increase profitability, modernizing the organization and streamlining processes has been a priority in recent years. Furthermore, it was found that the DCs' layouts had not been revised in years, which focused the thesis on analyzing flow and layout impacts on the KPI efficiency; where increasing DC size was directly associated with decreasing efficiency. This lead to three sub-questions, which will be answered chronologically:

How does the internal flow affect efficiency and flexibility in PNL DC operations?

It was determined, that in order to increase efficiency, the travel- and handling costs should be reduced. By focusing on the outdated layout, it was found the dock door allocation was fixed, which conflicts with fluctuating demand that is changing from day to day. Thus, distances from inbound demand dock doors are the main factor determining traversed distance within the DC, and with a fixed dock door allocation, the shortest traversed distance could not be ensured.

Thus, it can be concluded that a fixed dock door allocation impedes efficiency by not considering demand patterns and distances, and increased efficiency can be achieved in PNL DC operations.

How can the DC operations be improved with regard to efficiency and flexibility?

It was established that by applying a demand-based dock door allocation setup, where distances are accounted for, a reduction in traversed distance and increased efficiency could be achieved. To quantify and confirm this postulation, a mathematical model was constructed; where the demand groups with the greatest demand were allocated to the dock doors with the shortest distance. The model only utilized the dock doors that are currently used for operations. The results found that 1,366 Km could be saved. Assuming an average forklift speed of 5 km/h, it would result in a saving of 230.2 working hours per year. Therefore, it was chosen to further explore this opportunity through a simulation model, which included an As-Is benchmark, a port allocation solution, and lastly, a solution where the DK91 Preliminary Sorting zone and dock doors were moved. The results indicated that the greatest impact was to relocate the dock doors. However, by moving the DK91 storage zone, a further decrease in the distance could be attained. It was found that the distance could be reduced by approximately 10% and a total time saving of 84 hours a year.

Lastly, it was recommended to implement scenario three which includes allocating the demand groups with the greatest demand at the nearest dock doors and moving the DK91 storage zone closer to the inbound dock doors.

It can be concluded, that by implementing a dynamic dock door allocation the PNL operations would reduce traversed distances and increase efficiency.

How can an alternative to the current state be implemented?

By considering the findings throughout the analysis, it was established, that to implement a dynamic dock door allocation setup the system would require real-time data. However, this is an extensive system to implement, which requires supply chain collaboration, improving data quality, and altering or enhancing existing systems. Therefore, the solution proposed was to implement the demand-based dock door allocation setup by predicting demand groups by forecasting; and gradually as maturity and supply chain visibility is improved, utilizing real-time data to ensure increased efficiency by reducing traversed distances.

Hence, the alternative to the current state, namely a demand-based dock door allocation setup, should be implemented gradually in line with progress toward data and process maturity along with SCV.

By analyzing the DK91 DC in Aalborg, how can the efficiency and flexibility of PNL DCs be improved?

Thus, the problem statement can be answered, as it is concluded that the reductions achieved by implementing a dynamic dock door allocation setup would improve operational efficiency and flexibility, which can be further improved by a greater focus on SCV and applying real-time data.

Bibliography

Altıok, T. and Melamed, B. [2010], *Simulation modeling and analysis with Arena*, Elsevier Acad. Press.

Armstrong, J. S. [2002], *Principles of forecasting: A handbook for researchers and Practitioners*, Kluwer academic.

Assad Mohammed [2023], 'Meetings with assad mohammed'.

Baak, M., Koopman, R., Snoek, H. and Klous, S. [2020], 'A new correlation coefficient between categorical, ordinal and interval variables with pearson characteristics', *Computational Statistics & Data Analysis* **152**, 107043.

Bartholdi, J. J. and Hackman, S. T. [2019], *Warehouse & Distribution Science: Release 0.98.1*, 0.98.1 edn, Georgia Institute of Technology.

BIRTA, L. G. [2021], *Modelling and simulation: Exploring dynamic system behaviour*, SPRINGER.

Blume, T. [2023], 'Lazyprophet v0.3.8 - github'.
URL: *https://github.com/tblume1992/LazyProphet*

Cai, L. and Zhu, Y. [2015], 'The challenges of data quality and data quality assessment in the big data era', *Data Science Journal* **14**(0), 2.
URL: *https://datascience.codata.org/articles/10.5334/dsj-2015-002/*

Clark, W. A. and Avery, K. L. [2010], 'The effects of data aggregation in statistical analysis', *Geographical Analysis* **8**(4), 428–438.

Croston, J. D. [1972], 'Forecasting and stock control for intermittent demands', *Operational Research Quarterly (1970-1977)* **23**(3), 289.

Cumming, G. [2013], 'The new statistics', *Psychological Science* **25**(1), 7–29.

Dyckhoff, H., Derigs, U., Salomon, M., Tijms, H. C., Giesberts, P. and Wijngaard, J. [1994], Aggregation and disaggregation in demand forecasting, *in* 'Operations Research Proceedings 1993: DGOR/NSOR Papers of the 22nd Annual Meeting of DGOR in

Cooperation with NSOR/Vorträge der 22. Jahrestagung der DGOR zusammen mit NSOR', Springer, pp. 328–328.

Friedman, L. and Komogortsev, O. V. [2019], 'Assessment of the effectiveness of seven biometric feature normalization techniques', *IEEE Transactions on Information Forensics and Security* **14**(10), 2528–2536.

Gates, M. [2021], '6 simple ways to make your forklifts safer'.
URL: *https://www.tmhnc.com/blog/forklift-safety-accessories-and-osha-compliance*

GuolinKe, Q. M., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.-Y. [2017], 'Lightgbm: A highly efficient gradient boosting decision tree', *Adv. Neural Inf. Process. Syst* **30**, 52.

Hackman, S. T., Frazelle, E. H., Griffin, P. M., Griffin, S. O. and Vlasta, i. A. [2001], 'Benchmarking warehousing and distribution operations: an input-output approach', *Journal of Productivity Analysis* **16**, 79–100.

Hoad, K., Robinson, S. and Davies, R. [2010], 'Automated selection of the number of replications for a discrete-event simulation', *Journal of the Operational Research Society* **61**(11), 1632–1644.

Hyndman, R. J. and Athanasopoulos, G. [2023], *Forecasting: Principles and Practice*, 3 edn, OTexts.

Hyndman, R. J. and Khandakar, Y. [2008], 'Automatic time series forecasting: the forecast package for r', *Journal of statistical software* **27**, 1–22.

Hyndman, R. J. and Koehler, A. B. [2006], 'Another look at measures of forecast accuracy', *International Journal of Forecasting* **22**(4), 679–688.

Kalaiarasan, R., Agrawal, T. K., Olhager, J., Wiktorsson, M. and Hauge, J. B. [2022], 'Supply chain visibility for improving inbound logistics: A design science approach', *International Journal of Production Research* p. 1–16.

Kowalski, C. J. [1972], 'On the effects of non-normality on the distribution of the sample product-moment correlation coefficient', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **21**(1), 1–12.

Kwiatkowski, D., Phillips, P. C., Schmidt, P. and Shin, Y. [1992], 'Testing the null hypothesis of stationarity against the alternative of a unit root: How sure are we that economic time series have a unit root?', *Journal of Econometrics* **54**(1), 159–178.
URL: *https://www.sciencedirect.com/science/article/pii/030440769290104Y*

Law, A. M. [2015], *Simulation modeling and analysis*, 5 edn, McGraw-Hill.

Lei, M., Yin, Z., Li, S. and Tan, Q. [2017], Intermittent demand forecasting and inventory control with multiple temporal and cross-sectional aggregation and disaggregation methods, *in* '2017 13th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)', IEEE, pp. 1969–1977.

Linde [2022], 'Linde t20-25sp/sp b'.
URL: *https://www.nc-nielsen.dk/produkt/linde-t20-25sp-sp-b?fbclid=IwAR0Z29exh73Fvtc90VoRrYzDSkjNzXNpnca0lA7BFsJIRdAICPEBp-0HqsM*

Novy-Marx, R. [2013], 'The other side of value: The gross profitability premium', *Journal of financial economics* **108**(1), 1–28.

PostNord [2022], 'History'.
URL: *https://www.postnord.com/about-us/history*

Postnord [2023], 'Third-party logistics (tpl)'.
URL: *https://www.postnord.dk/en/our-solutions/tpl*

Premkumar, G., Ramamurthy, K. and Nilakanta, S. [1994], 'Implementation of electronic data interchange: an innovation diffusion perspective', *Journal of Management Information Systems* **11**(2), 157–186.

Ratner, B. [2009], 'The correlation coefficient: Its values range between +1/-1, or do they?', *Journal of Targeting, Measurement and Analysis for Marketing* **17**(2), 139–142.

Robinson, S. [1997], Simulation model verification and validation: increasing the users' confidence, *in* 'Proceedings of the 29th conference on Winter simulation', pp. 53–59.

Rožanec, J. M., Fortuna, B. and Mladenić, D. [2022], 'Reframing demand forecasting: A two-fold approach for lumpy and intermittent demand', *Sustainability* **14**(15), 9295.

Saranya, C. and Manikandan, G. [2013], 'A study on normalization techniques for privacy preserving data mining', *International Journal of Engineering and Technology (IJET)* **5**(3), 2701–2704.

Schuessler, A. A. [1999], 'Ecological inference', *Proceedings of the National Academy of Sciences* **96**(19), 10578–10581.

SciPy [2023*a*], 'Scipy stats: Calculate a spearman correlation coefficient with associated p-value.'.
URL: *https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html*

SciPy [2023*b*], 'Scipy stats: Calculate kendall's tau, a correlation measure for ordinal data.'.
URL: *https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html*

SciPy [2023*c*], 'Scipy stats: Pearson correlation coefficient and p-value for testing non-correlation'.
URL: *https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html*

Sharma, D. K., Singh, B., Raja, M., Regin, R. and Rajest, S. S. [2021], 'An efficient python approach for simulation of poisson distribution', *2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS)* .

Sim, S. [2000], Next generation data interchange: tool-to-tool application program interfaces, *in* 'Proceedings Seventh Working Conference on Reverse Engineering', pp. 278–280.

Smithson, M. and Popovich, P. M. [2003], *Correlation: Parametric and nonparametric measures*, Sage Publications.

Sumah, B., Masudin, I., Zulfikarijah, F. and Restuputri, D. P. [2020], 'Logistics management and electronic data interchange effects on logistics service providers' competitive advantage', *Journal of Business and Economic Analysis* **3**(02), 171–194.

Syntetos, A. and Boylan, J. [2001], 'On the bias of intermittent demand estimates', *International Journal of Production Economics* **71**(1–3), 457–466.

Taylor, S. J. and Letham, B. [2018], 'Forecasting at scale', *The American Statistician* **72**(1), 37–45.

VIRK [2022], 'Postnord logistics a/s'.
URL: *https://datacvr.virk.dk/enhed/virksomhed/20148586?fritekst=20148586&sideIndex=0&size=10*

Witte, R. S. and Witte, J. S. [2019], *Statistics*, Wiley.

Wright, S. [1921], 'Correlation and causation', *Journal of Agricultural Research* **20**(7), 557–585.

Zwillinger, D. and Kokoska, S. [2000], *Standard probability and statistics tables and formulae*, Chapman and Hall/CRC.

A Dock Door

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
12	22,8	18988	438	332	432926,4	9986,4	7569,6
10	32,8	17289	374	324	567079,2	12267,2	10627,2
7	47,2	16894	288	250	797396,8	13593,6	11800
11	27,7	15981	348	225	442673,7	9639,6	6232,5
9	37,2	15315	309	288	569718	11494,8	10713,6
26	51,6	13247	0	144	683545,2	0	7430,4
8	42,6	13143	286	196	559891,8	12183,6	8349,6
22	32,6	11943	183	183	389341,8	5965,8	5965,8
18	14,1	10834	251	181	152759,4	3539,1	2552,1
13	18,4	8686	215	137	159822,4	3956	2520,8
15	9,2	6010	130	0	55292	1196	0
14	14	4684	97	95	65576	1358	1330
25	47,1	2661	51	42	125333,1	2402,1	1978,2

Figure A.1. Distance. demand and accumulated distance filtered on demand for 1 year
Spread Flow

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
12	22,8	18988	438	332	432926,4	9986,4	7569,6
10	32,8	17289	374	324	567079,2	12267,2	10627,2
11	27,7	15981	348	225	442673,7	9639,6	6232,5
9	37,2	15315	309	288	569718	11494,8	10713,6
7	47,2	16894	288	250	797396,8	13593,6	11800
8	42,6	13143	286	196	559891,8	12183,6	8349,6
18	14,1	10834	251	181	152759,4	3539,1	2552,1
13	18,4	8686	215	137	159822,4	3956	2520,8
22	32,6	11943	183	183	389341,8	5965,8	5965,8
15	9,2	6010	130	0	55292	1196	0
14	14	4684	97	95	65576	1358	1330
25	47,1	2661	51	42	125333,1	2402,1	1978,2
26	51,6	13247	0	144	683545,2	0	7430,4

Figure A.2. Distance. demand and accumulated distance filtered on demand for 1 week in June
Spread Flow

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
12	22,8	18988	438	332	432926,4	9986,4	7569,6
10	32,8	17289	374	324	567079,2	12267,2	10627,2
9	37,2	15315	309	288	569718	11494,8	10713,6
7	47,2	16894	288	250	797396,8	13593,6	11800
11	27,7	15981	348	225	442673,7	9639,6	6232,5
8	42,6	13143	286	196	559891,8	12183,6	8349,6
22	32,6	11943	183	183	389341,8	5965,8	5965,8
18	14,1	10834	251	181	152759,4	3539,1	2552,1
26	51,6	13247	0	144	683545,2	0	7430,4
13	18,4	8686	215	137	159822,4	3956	2520,8
14	14	4684	97	95	65576	1358	1330
25	47,1	2661	51	42	125333,1	2402,1	1978,2
15	9,2	6010	130	0	55292	1196	0

Figure A.3. Distance. demand and accumulated distance filtered on demand for one week in December
Spread Flow

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
6	8,4	12858	158	172	108007,2	1327,2	1444,8
17	4,65	10621	139	143	49387,65	646,35	664,95
12	16,1	819	0	0	13185,9	0	0
11	12,7	775	0	0	9842,5	0	0
10	17,5	737	0	0	12897,5	0	0
9	22,5	663	12	33	14917,5	270	742,5
8	27	488	0	0	13176	0	0
22	5,3	335	6	11	1775,5	31,8	58,3
18	25,6	291	4	16	7449,6	102,4	409,6
7	31,8	278	3	10	8840,4	95,4	318
13	22,5	271	0	0	6097,5	0	0
26	26,1	200	0	0	5220	0	0
14	27	184	0	14	4968	0	378
15	31,8	164	0	0	5215,2	0	0
25	21,5	51	2	2	1096,5	43	43

Figure A.4. Distance. demand and accumulated distance filtered on demand for 1 year Collection Flow

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
6	8,4	12858	158	172	108007,2	1327,2	1444,8
17	4,65	10621	139	143	49387,65	646,35	664,95
9	22,5	663	12	33	14917,5	270	742,5
22	5,3	335	6	11	1775,5	31,8	58,3
18	25,6	291	4	16	7449,6	102,4	409,6
7	31,8	278	3	10	8840,4	95,4	318
25	21,5	51	2	2	1096,5	43	43
12	16,1	819	0	0	13185,9	0	0
11	12,7	775	0	0	9842,5	0	0
10	17,5	737	0	0	12897,5	0	0
8	27	488	0	0	13176	0	0
13	22,5	271	0	0	6097,5	0	0
26	26,1	200	0	0	5220	0	0
14	27	184	0	14	4968	0	378
15	31,8	164	0	0	5215,2	0	0

Figure A.5. Distance. demand and accumulated distance filtered on demand for 1 week in June Collection Flow

Dock Door	Distance	Demand 1 Year	Demand for June	Demand for December	Total Distance 1 Year	Total Distance June	Total Distance December
6	8,4	12858	158	172	108007,2	1327,2	1444,8
17	4,65	10621	139	143	49387,65	646,35	664,95
9	22,5	663	12	33	14917,5	270	742,5
18	25,6	291	4	16	7449,6	102,4	409,6
14	27	184	0	14	4968	0	378
22	5,3	335	6	11	1775,5	31,8	58,3
7	31,8	278	3	10	8840,4	95,4	318
25	21,5	51	2	2	1096,5	43	43
12	16,1	819	0	0	13185,9	0	0
11	12,7	775	0	0	9842,5	0	0
10	17,5	737	0	0	12897,5	0	0
8	27	488	0	0	13176	0	0
13	22,5	271	0	0	6097,5	0	0
26	26,1	200	0	0	5220	0	0
15	31,8	164	0	0	5215,2	0	0

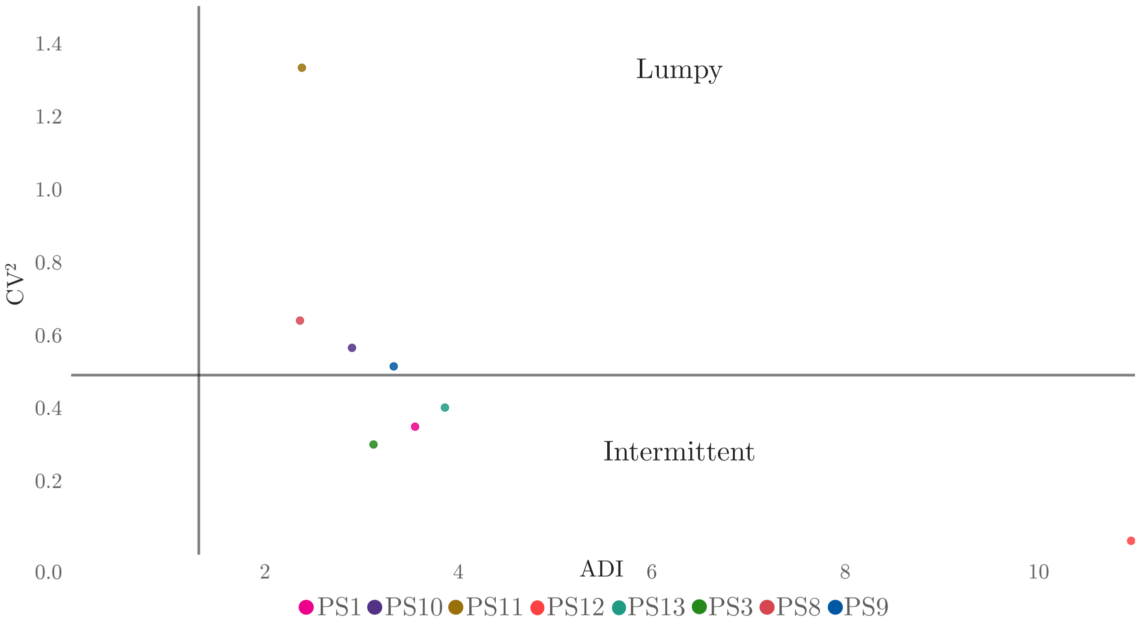
Figure A.6. Distance. demand and accumulated distance filtered on demand for one week in December
Collection Flow

B Postal Codes

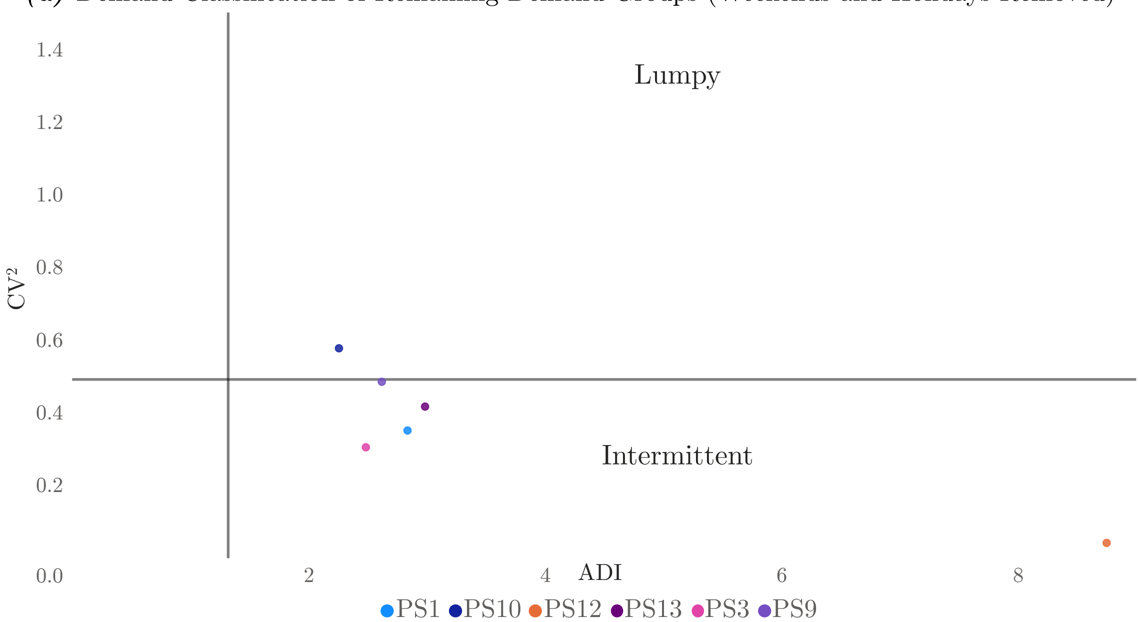
Abbreviations	Postal Codes
DK37	3080,3100,3120,3140,3150,3200,3230,3330,3390,3470,3500,3520,3540,3550,3600.3630.3660, 3670,3700,3720,3770,3782,3790
DK48	4000,4030,4040,4050,4060,4070,4100,4130,4140,4160,4171,4174,4180,4190,4200,4220,4230, 4241,4242,4243,4276,4281,4296,4300,4305,4310,4330,4340,4350,4360,4376,4390,4400,4420, 4440,4450,4460,4470,4480,4490,4500,4520,4532,4534,4540,4550,4560,4571,4572,4581,4583, 4591,4593,4600,4621,4622,4623,4653,4654,4660,4671,4672,4673,4681,4682,4683,4684,4690, 4700,4720,4733,4771,4472,4780,4791,4792,4793,4800,4850,4862,4863,4874,4872,4873
DK50	5000,5100,5200,5210,5220,5230,5240,5250,5260,5270,5290,5300,5320,5230,5250,5370,5380, 5290,5400,5450,5462,5463,5464,5466,5471,5474,5485,5491,5500,5540,5550,5560,5580,5591, 5592,5600,5603,5610,5632,5642,5672,5683,5700,5750,5762,5771,5772,5792,5800,5853,5856, 5871,5881,5882,5883,5884,5900,5932,5935,5960
DK69	6000,6040,6051,6052,6092,6093,6094,6100,6200,6230,6240,6261,6270,6320,6330,6340,6360, 6372,6392,6400,6430,6440,6470,6500,6510,6520,6534,6535,6541,6560,6580,6600,6621,6622, 6623,6630,6640,6650,6660,6670,6682,6683,6690,6700,6765,6705,6710,6715,6720,6731,6740, 6752,6753,6760,6771,6780,6793,6800,6816,6823,6830,6840,6851,6900,6920,6933,6940,6950, 6960,6873,6980,6990
DK73	7000,7007,7018,7080,7100,7120,7130,7140,7150,7160,7173,7171,7182,7183,7184,7190,7200, 7250,7260,7280,7300,7321,7323,7330,7361,7362,7400,7430,7441,7442,7451,7470,7480,7496, 7500,7550,7560,7570,7600,7620,7650,7669,7673,7680,7700,7730,7741,7742,7752,7755,7760, 7770,7790,7800,7830,7840,7850,7860,7870,7884,97900,7950,7960,7980,7990
DK81	8100,8200,8230,8330,8240,8245,8250,8260,8270,8300,8310,8320,8330,8340,8350,8355,8361, 8362,8370,8380,8381,8382,8400,8410,8420,8444,8450,8462,8464,8471,8500,8520,8530,8541, 8543,8544,8550,8560,8570,8581,8585,8586,8592,8600,8620,8632,8641,8643,8670,8680,8700, 8721,8722,8723,8740,8751,8765,8781,8799,8800,8830,8831,8832,8870,8881,8882,8900,8920, 8940,8950,8990

Table B.1. Postal code overview for DK37/48/50/69/73/81

Demand Classification C



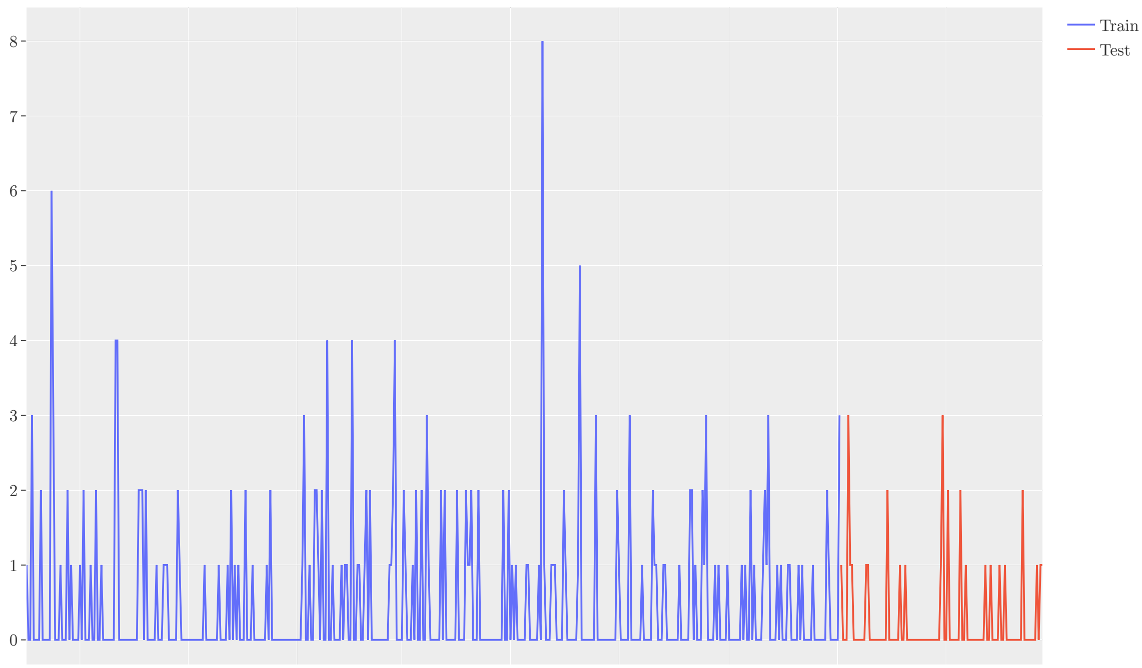
(a) Demand Classification of Remaining Demand Groups (Weekends and Holidays Removed)



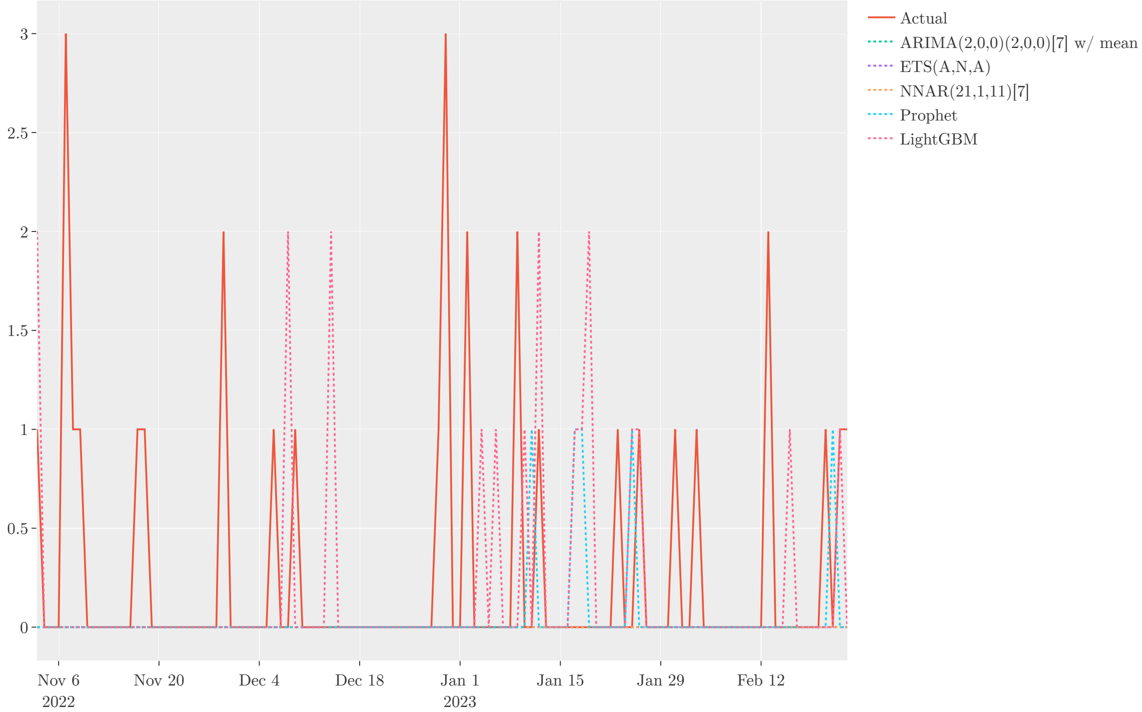
(b) Demand Classification of Remaining Demand Groups (Weekends and Holidays Removed)

Figure C.1. Demand Classification Graphs of Forecastability (2021-05-07 - 2023-03-01)

Extra Forecast Plots D



Predicted values PS13



(b) PS-13 Demand - Prediction (Intermittent)

Figure D.1. Train Test splits from Forecasts)

Code E

This chapter includes all code scripts written for this thesis. Primarily, code has been written in Python with use of common libraries such as `pandas` and `numpy`. When other libraries are used, this will be explicitly stated. In total, around 550 lines (≈30,000 characters) of code has been used for data manipulation and visualization in this thesis. An overview, explanations and descriptions will be provided throughout this chapter.

E.1 Code for Correlation Analysis

This code relates to the creation of boxplots and code for normalizing and calculating the correlation values for the KPI scan throughput in Section [2.1].

```
#Import Terminal Data
dfKPI = pd.read_excel(r'KPI Terminal Produktivitet.xlsx', sheet_name="Clean", index_col=False)
#Sort Dataframe to Control Boxplot Visual Order
sortlst = []
for Terminal in dfKPI.Terminal:
    if Terminal == "Aalborg":
        sortlst.append(1)
    if Terminal == "Herning":
        sortlst.append(2)
    if Terminal == "Aarhus":
        sortlst.append(3)
    if Terminal == "Taulov":
        sortlst.append(4)
    if Terminal == "Køge":
        sortlst.append(5)
dfKPI['SortCol'] = sortlst
dfKPI.sort_values(by=['SortCol'], inplace=True)
#Create boxplot
fig = px.box(KPI, x="Terminal", y="KPI", points="all", hover_data=["Uge", "Year"], color="Terminal", width=1450, height=920)
fig.show()
fig.write_image("boxplot.pdf")
```

Listing E.1. Correlation: Scan Througput KPI Boxplot

In Listing [E.1], the code provided simply imports raw scan throughput data, sorts it according to the respective distribution center and appropriates it for a boxplot graph.

```
34 #Import Data Containing Demand and KPIs
35 df = pd.read_excel(r'KPI Terminal Produktivitet.xlsx', sheet_name="DK91DemandAndKPI",
       index_col=False)
37 #Function for Min-Max Normalization
38 def MinMax(x):
39     return (x - min(x))/(max(x) - min(x))
41 #Function for Median & Median Absolute Deviation (MMAD) Normalization
42 def MMAD(x):
43     def MAD(y):
44         return median(abs(y - median(y)))
45     return (median(x) - x)/MAD(x)
47 methodlst = [MinMax, MMAD]
48 for idx, norm in enumerate(methodlst):
49     df['KPINorm'], df['DemandNorm'] = norm(df.KPI), MinMax(df.Demand)
50     x, y = df['KPINorm'], df['DemandNorm']
51     if idx == 1:
52         print("MinMax Normalization")
53     else:
54         print("MMAD Normalization")
55     print("Pearson's r", pearsonr(x, y))
56     print("Kendall's Tau", kendalltau(x, y))
57     x, y = df[['KPINorm', 'DemandNorm']].T.values
58     print('phik         = %.2f'%phik.phik_from_array(x, y, num_vars=['x']))
```

Listing E.2. Correlation: Analysis Between DK91 Demand and Scan Throughput

In Listing [E.2], the same excel sheet is imported, but a different sheet which includes demand for DK91 and the relevant KPI's. In order to compare the the KPI to the demand, it was necessary to rescale these, as comparing two variables with different scales and units would distort the analysis. From lines 37-47, two types of normalization were implemented and tested, however they provided the same results and it was chosen to only apply min-max normalization for scaling the data. The remaining lines simply normalize the imported data columns containing the DK91 KPI's and the demand in the same period of DK91. Then, the correlation coefficients for Pearson, Kendall and ϕ_K are calculated. These commands are achieved with the use of `scipy.stats` and `phik` libraries.

E.2 Code for Simulation Arrival Tables

For the code presented in this section, it is written with the purpose of producing an output that can be used directly for simulation, relating to the arrival tables and thereby the logic of demand flow in the DK91 distribution center. Thus, to make it simple to use, it has been implemented as a function to allow multiple set-ups and scenarios such as changing date ranges and desired setup with a one-liner.

```
12 def DoorAllocation(df, GSAsIs=True, DoorAllocationAsIs=True):
13     import warnings
14     warnings.simplefilter(action='ignore', category=FutureWarning)
15     import pandas as pd
16     pd.options.mode.chained_assignment = None
17     from numpy import ceil, nan
21     DK91grouplst = [["9990", "9981", "9982", "9970", "9870", "9881"], ...
31                     ]
32     holidaylst = ["2021-01-01", "2021-04-01", "2021-04-02", "2021-04-05", "2021-04-30", "2021-05-13", "2021-05-24", ...
36                   ]
38     temp1 = df[(df.shipment_department == 'DK91') & (df.delivery_department == 'DK91')]  # GS DK91 Filter
39     temp2 = df[(df.shipment_department == 'DK91') & (df.delivery_department.isin(['DK81', 'DK69']))]  # GS DK81/69 Filter
40     temp3 = df[(df.shipment_department == 'DK91') & (df.delivery_department != 'DK91')]  # GS Rest Filter
41     temp4 = df[(df.shipment_department != 'DK91') & (df.delivery_department == 'DK91')]  # FS Filter
```

Listing E.3. Simulation: Arrival Table Function (1)

For the first snippet in Listing [E.3], where the function is introduced, the libraries are imported with the function: `warnings`, `pandas` and `numpy`, where the first library simply is used to suppress a certain warning occurring in the code terminal for clearer output. In lines 21-36, two lists are introduced, *DK91Grouplst* and *holidaylst*. The first nested list contains each postal code for each of the 13 demand groups in the DK91 area. The other list contains each non-weekend holiday from 2021-2023 roughly. The *holidaylst* will later be used to remove demand in holidays and weekend-days, as there is no production on these dates.

For the last lines in the snippet, four dataframes are defined. Each dataframe is filtered and defined by the given demand groups, either preliminary sorting (GS), which is split into three subgroups DK91, DK81/DK69 and the remaining, reffered to as *GS-Rest*; and lastly, secondary sorting (FS).

```
43  #GSDK81/69 and GSRest
44  Restgrouplst = [temp2, temp3]
45  Restgrouplst2 = []
46  for idx, tempdf in enumerate(Restgrouplst):
47      Df = tempdf.set_index(tempdf.kp2_datetime)
48      Df = Df.resample('H').sum().fillna(0)
49      Df['Date'] = Df.index
50      Df['Weekday'] = Df.Date.dt.weekday + 1
51      Df = Df[Df.Weekday < 6]
52      Df.Date = Df.index.strftime('%Y-%m-%d')
53      Df = Df[~(Df.Date.isin(holidaylst))]
54      Df.index = Df.index.strftime('%H:%M:%S')
55      Df = (Df.drop(columns=['Weekday']).groupby(['kp2_datetime']).mean()).reset_index()
     #.apply(ceil)
56      Df.kolli[7:14] = [x + y for x, y in zip(Df.kolli[0:7], Df.kolli[7:14])]
57      Df.kolli[14:21] = [x + y for x, y in zip(Df.kolli[7:14], Df.kolli[14:21])]
58      Df.kolli[18:21] = [x + y for x, y in zip(Df.kolli[18:21], Df.kolli[21:24])]
59      Df.kolli[0:14], Df.kolli[21:24] = 0, 0
60      Df['Grovdest'] = int(idx + 2)
61      Df.kolli = Df.kolli.apply(ceil)
62      Df['Group'] = str(idx + 14)
63      Df['dest'] = nan
64      Df['QtySum'] = Df.kolli.sum()
65      Df.kp2_datetime = ["hr(" + str(h) + ")" for h in range(24)]
66      Restgrouplst2.append(Df)
```

Listing E.4. Simulation: Arrival Table Function (2)

Due to the nature of the different demand groups, they will also be treated differently. Specifically, the *GSDK81/69* and *GSRest* groups are a 'complete' group, meaning they are not split into subgroups, but a treated directly. The procedure the groups go through are virtually the same, where each group has to appropriated in 24-hour time bins, referred to as resampling, and consequently populating the demand in the operating hours of the specific group.

For starters in Listing [E.4], the two demand groups *GSDK81/69* and *GSRest* are put in a list, such that they can be iterated through. The results of the iteration will be stored in the list *Restgrouplst2*. On line 46-55 the loop iterates through two variables *idx* and *tempdf* in the list previosuly created containing the two dataframes. The two iterators are for keeping track of the current iteration and the latter representing a dataframe in the list. Firstly, the column containing scan times is resampled into hourly bins, and to ensure weekdays and holidays are excluded, the *Df['WeekDay']* column represents each day of a week numerically from 1-7, where all weekend-days are filtered out in line 51; the same principle is applied for holidays, with dates from *holidaylst* are removed. On lines 54-55, all dates and hours are consolidated into 24-hour bins, meaning an average of all demands in the hourly bins from 0-23 is put into their respective hour.

From line 56-59 the demand is shifted into operating hours exclusively which is 14:00-20:00 for these groups. The remaining lines 60-66 is about labelling and re-purposing the data into the correct formats required for the simulation arrival tables; since the logic is controlled by labels in the simulation model. Here, the *idx* iterator is used to provide the correct labels for the simulation model as well as provide ID for further use of the code. Lastly, the iterated dataframe is appended and stored in a list.

```
68 #GSDK91 Groups
69 GSDK91dflst = []
70 for idx, group in enumerate(DK91grouplst):
71     Df = temp1[temp1['consignee_place_code'].isin(group)]
72     Df.set_index(Df.kp2_datetime, inplace=True)
73     Df = Df.resample('H').sum().fillna(0)
74     Df['Date'] = Df.index
75     Df['Weekday'] = Df.Date.dt.weekday + 1
76     Df = Df[Df.Weekday < 6]
77     Df.Date = Df.index.strftime('%Y-%m-%d')
78     Df = Df[~(Df.Date.isin(holidaylst))]
79     Df.index = Df.index.strftime('%H:%M:%S')
80     Df = (Df.drop(columns=['Weekday']).groupby(['kp2_datetime']).mean()).reset_index()
 #.apply(ceil)
81     Df.kolli[7:14] = [x + y for x, y in zip(Df.kolli[0:7], Df.kolli[7:14])]
82     Df.kolli[14:21] = [x + y for x, y in zip(Df.kolli[7:14], Df.kolli[14:21])]
83     Df.kolli[18:21] = [x + y for x, y in zip(Df.kolli[18:21], Df.kolli[21:24])]
84     Df.kolli[0:14], Df.kolli[21:24] = 0, 0
85     Df.kolli = Df.kolli.apply(ceil)
86     Df['QtySum'] = Df.kolli.sum()
87     Df['Group'] = str(idx + 1)
88     Df.kp2_datetime = ["hr(" + str(h) + ")" for h in range(24)]
89     GSDK91dflst.append(Df)
```

Listing E.5. Simulation: Arrival Table Function (3)

The code snippet in Listing [E.5] is similar to the code provided in Listing [E.4]. The difference is, the list iterated through, *DK91grouplst*, is the demand group list containing postal codes for each of the 13 groups defined in Listing [E.3]. Here, each iteration filters the dataframe by postal codes contained in each group. The working hours remain the same and are thus ’shifted’ and populated in the same 14:00-20:00 bins; and each group is stored and identified by the created *Df[’Group’]* column. Lastly, the dataframes are stored in the list, *GSDK91dflst*.

```
#FSDK91 Groups
FSDK91dflst = []
for idx, group in enumerate(DK91grouplst):
    Df = temp4[temp4[’consignee_place_code’].isin(group)]
    Df.set_index(Df.kp2_datetime, inplace=True)
    Df = Df.resample(’H’).sum().fillna(0)
    Df[’Date’] = Df.index
    Df[’Weekday’] = Df.Date.dt.weekday + 1
    Df = Df[Df.Weekday < 6]
    Df.Date = Df.index.strftime(’%Y-%m-%d’)
    Df = Df[~(Df.Date.isin(holidaylst))]
    Df.index = Df.index.strftime(’%H:%M:%S’)
    Df = (Df.drop(columns=[’Weekday’]).groupby([’kp2_datetime’]).mean()).reset_index()
 #.apply(ceil)
    Df.kolli[7:15] = [x + y for x, y in zip(Df.kolli[15:23], Df.kolli[7:15])]
    Df.kolli[23] = Df.kolli[23] + Df.kolli[7]
    Df.kolli[1:7] = [x + y for x, y in zip(Df.kolli[1:7], Df.kolli[8:15])]
    Df.kolli[7:23] = 0
    Df.kolli = Df.kolli.apply(ceil)
    Df[’QtySum’] = Df.kolli.sum()
    Df[’Group’] = str(idx + 1)
    Df.kp2_datetime = ["hr(" + str(h) + ")" for h in range(24)]
    FSDK91dflst.append(Df)
```

Listing E.6. Simulation: Arrival Table Function (4)

Treating the last group of secondary sorting, in Listing [E.6] we see a near identical code compared to Listing [E.5]. The logic is the same, however, the key difference being the operating hours of the secondary shorting team, 23:00-06:00. This means shifting demand into the relevant time bins is different from the time bins in Listings [E.4] and [E.5]. Otherwise, the same procedure is applied and the dataframes are stored in *FSDK91dflst*.

```
#Scenario Selection
if GSAsIs and DoorAllocationAsIs:
    GSDK91dfconcat = pd.concat(GSDK91dflst).reset_index(drop=True)
    FSDK91dfconcat = pd.concat(FSDK91dflst).reset_index(drop=True)
    Doorlst = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 14, 17, 18]  # As-Is GS, As-Is Door Allocation
    returnpath = 'ED_DoorAllocation_' + 'GS1_' + 'DA1'
#Dock Door Ordering for Allocation
if (GSAsIs and not DoorAllocationAsIs):
    Doorlst = [2, 11, 3, 12, 4, 13, 5, 15, 6, 16, 7, 17, 8]  # As-Is GS, To-Be Door Allocation
    returnpath = 'ED_DoorAllocation_' + 'GS1_' + 'DA2'
    GSDK91dfconcat = pd.concat(GSDK91dflst).sort_values(by=['QtySum', 'Group'], ascending=False).reset_index(drop=True)
    FSDK91dfconcat = pd.concat(FSDK91dflst).sort_values(by=['QtySum', 'Group'], ascending=False).reset_index(drop=True)
if (not GSAsIs and not DoorAllocationAsIs):
    Doorlst = [13, 4, 12, 3, 14, 5, 11, 2, 15, 6, 16, 7, 17]  # To-Be GS, To-Be Door Allocation
    returnpath = 'ED_DoorAllocation_' + 'GS2_' + 'DA2'
    GSDK91dfconcat = pd.concat(GSDK91dflst).sort_values(by=['QtySum', 'Group'], ascending=False).reset_index(drop=True)
    FSDK91dfconcat = pd.concat(FSDK91dflst).sort_values(by=['QtySum', 'Group'], ascending=False).reset_index(drop=True)
#GSDK91 Dock Door Allocation
GSDK91dflst2, FSDK91dflst2 = [], []
for group, door in zip(GSDK91dfconcat.Group.unique(), Doorlst):
    GSDK91temp = GSDK91dfconcat[GSDK91dfconcat.Group == group]
    GSDK91temp['dest'], GSDK91temp['Grovdest'] = door, 1
    GSDK91dflst2.append(GSDK91temp)
#FSDK91 Dock Door Allocation
for group, door in zip(FSDK91dfconcat.Group.unique(), Doorlst):
    FSDK91temp = FSDK91dfconcat[FSDK91dfconcat.Group == group]
    FSDK91temp['dest'], FSDK91temp['Grovdest'] = door, nan
    FSDK91dflst2.append(FSDK91temp)
```

Listing E.7. Simulation: Arrival Table Function (5)

The long snippet in Listing [E.7] pertains to allocating each demand group depending on the function input. By measuring the distances to each port, we have manually stored the correct order of dock doors, relative to different scenarios. As-Is mirrors the current setup at PNL's DC, but the To-Be is always the shortest distance to any given dock door from the origin point of preliminary sorting. Thus, each preliminary sorting group is sorted according to highest demand and dock door labels are assigned accordingly in a descending order: Highest demand is assigned the next label in the list. This allows the following scenario setups:

1. **As-Is** ($GSAsIS$ = `True`, $DoorAllocationAsIS$ = `True`):

   This scenario is, as the name suggests, a mirror of PNL's DK91 DC current operations. Preliminary sorting is thus as-is along with the dock door allocation.
2. **To-Be Dock Door Allocation** ($GSAsIS$ = `True`, $DoorAllocationAsIS$ = `False`):

   In this scenario, dock door allocation is decided by the given demand (highest demand) and the shortest distance from the preliminary sorting point. The preliminary sorting point is unchanged from the current operations.
3. **To-Be Operations** ($GSAsIS$ = `False`, $DoorAllocationAsIS$ = `False`):

   With the last scenario, the entire setup is challenged by moving the preliminary sorting point and allocation of dock doors is given from this new placement.

This logic given above, is written in lines 116-137. The remaining lines allocates dock doors by label according to the one of the given scenario setup choices, given as an input to the function. This is applied to both the PS-DK91 and SS groups, where the results are appended and stored in lists.

```
153 GSdf = pd.concat(GSDK91dflst2 + Restgrouplst2)
154 FSdf = pd.concat(FSDK91dflst2)
155 Alldfs = []
156 for dfnumber, dfs in enumerate([GSdf, FSdf]):
157     dfs['Channel'] = 1
158     dfs['AtomName'] = "A"
159     dfs = dfs[['kp2_datetime', 'AtomName', 'kolli', 'Channel', 'dest', 'Grovdest', 'Group', 'QtySum']]
160     dfs.columns = ['ArrivalTime', 'AtomName', 'Quantity', 'Channel', 'dest', 'Grovdest', 'Group', 'QtySum']
161     dfs = dfs[dfs.Quantity != 0].reset_index(drop=True)
162     dfs = dfs.sort_values(by=['ArrivalTime'], ascending=True)
163     dfs.Quantity = ['Poisson(' + str("%d" % x) + ')' if x > 0 else 0 for x in dfs.Quantity]
164     Alldfs.append(dfs)
166 with pd.ExcelWriter(returnpath + datepath + '.xlsx') as writer:
167     Alldfs[0].to_excel(writer, sheet_name='GS', index=False)
168     Alldfs[1].to_excel(writer, sheet_name='FS', index=False)
```

Listing E.8. Simulation: Arrival Table Function (6)

In Listing [E.8], the last snippet of this code, the dataframes containing each of the 28 groups are appropriated to fit the output of an applicable arrival table for the simulation model and output to an excel file, where the file name is determined by the function input, therefore not only determining the scenario setup, but also the name of the file - thus providing easy overview of output files. The excel-file is split into two sheets, one for preliminary sorting and one for secondary sorting. This concludes the code, and the data is sorted and appropriated, ready for use in the simulation model.

E.3 Code for Demand Classification and Forecast Input

This code is written to calculate ADI and CV^2 for demand classification of demand groups, as well as prepare and label data for forecasting.

```
tmplst1, tmplst2 = [temp1, temp4], [temp2, temp3]
dflst = []
for Idx, tempdf in enumerate(tmplst1):
    for idx, group in enumerate(DK91grouplst):
        Df = tempdf[tempdf['consignee_place_code'].isin(group)]
        Df.set_index(Df.kp2_datetime, inplace=True)
        Df = Df.resample('D').sum().fillna(0)
        Df['Date'] = Df.index
        Df = Df.rename(columns={'kolli': 'Demand'}).reset_index(drop=True)
        Df['Weekday'] = Df.Date.dt.weekday + 1
        Df = Df[Df.Weekday < 6]
        Df = Df[~(Df.Date.isin(holidaylst))].drop(columns=['Weekday'])
        Df['Group'] = idx+1
        if Idx == 0:
            Df['ID'] = 'GS'
        if Idx == 1:
            Df['ID'] = 'FS'
        dflst.append(Df[['Date', 'Demand', 'Group', 'ID']])
for Idx, tempdf in enumerate(tmplst2):
    Df = tempdf.set_index(tempdf.kp2_datetime)
    Df = Df.resample('D').sum().fillna(0)
    Df['Date'] = Df.index
    Df = Df.rename(columns={'kolli': 'Demand'}).reset_index(drop=True)
    Df['Weekday'] = Df.Date.dt.weekday + 1
    Df = Df[Df.Weekday < 6]
    Df = Df[~(Df.Date.isin(holidaylst))].drop(columns=['Weekday'])
    Df['Group'], Df['ID'] = Idx + 14, 'GS'
    # Df.rename(columns={'kolli': 'Demand'})
    dflst.append(Df[['Date', 'Demand', 'Group', 'ID']])
```

```
64 dfs = pd.concat(dflst).reset_index(drop=True)
65 dfs.to_excel('ForecastDemand(NS).xlsx', index=False)
```

Listing E.9. Demand Classification: Splitting Demand

For the first lines, they are identical to the code provided in Listing [E.3], which is intentional. In Listing [E.9], two lists contain the relevant dataframes and these two list are iterated through separately. It is essentially a more condensed version of the code written throughout Appendix [E.2]. The first lines 37-52 resamples the data, but not into hourly bins, but rather daily. This is due to the purpose of the data, is to be applied in time-series forecasting with a daily demand frequency. Preliminary sorting DK91 groups and secondary sorting are split into groups according to Fig. [4.9]. The groups are labeled with group number and demand group name, and finally appended to the list *dflst*. This equates to $13 \cdot 2 = 26$ dataframes containing each group, where the last two groups of DK81/69 and DKRest are handled in lines 53-65, with the same procedure and logic. The two remaining groups are dealt with, equalling 28 dataframes appended to *dflst*. In 64-65 the dataframes are concatenated and saved to an excel-file. This is this excel file will be used for further forecasting in Appendix [E.4].

E.4 Code for Forecasting

For this code, it is primarily written in R for its speed and simplicity advantage over Python. The applied data came from Appendix [E.3]. However, Python is also used to plot graphs and train a machine learning model for univariate time-series forecasting.

```r
library(fpp3)
library(readxl)
library(fable.prophet)
data <- read_excel("ForecastDemand.xlsx",sheet = "Sheet1") |>
  mutate(Date = as_date(Date)) |>
  as_tsibble(
    index = Date,
    key = c(Group, ID, gID)
  ) |>
  mutate(Demand_ihs = asinh(Demand))
```

Listing E.10. Forecasting R: Preparing Data

The first snippet in Listing [E.10], which is two separate chunks, loads the used libraries: `fpp3`, `readxl`, `fable.prophet`. The `fpp3` library is a forecasting library supplied with an online and free book on time-series forecasting by Hyndman and Athanasopoulos [2023]. This package has many useful functions, and includes three of the applied models, namely: *autoARIMA*, *ETS*, and *NNAR*. The last, more advanced, model is supplied by a standalone library from `fable`, only including functionality for the *Prophet* model. The `readxl` library allows reading and writing data from Excel-files.

From lines 8-14, the data is prepared by applying the *mutate* function in a pipe environment, and conforming the input data to a so called 'time-series *tibble*' or *tsibble*, which is essentially a dataframe for time series. The *key* argument allows multiple time-series to be stored in one tsibble. The key in our case is the 28 demand groups. On line 14, a column is added named *Demand_ ihs*, which is an inverse hyperbolic sine transformation (IHS) of the *Demand* column of our time-series. Time-series transformation are often used to improve automatic model selection, and the IHS transformation is useful for intermittent data, as it can handle non-positive values. This is unused in the thesis, as it didn't improve model and term selection.

```
29 ###Functions Definition###
30 p = 0.8 #Predefined size for test/train splits, 0.8 = 80/20
32 #Train split function
33 ftrain <- function(x, p) {
34   split <- round(nrow(x) * p)
35   train <- x[1:split, ]
36 }
37 #Test split function
38 ftest <- function(x, p) {
39   split <- round(nrow(x) * p)
40   test <- x[(split + 1):nrow(x), ]
41 }
43 ###Forecasting Loop###
44 lst <- list() #List for tsibble of test split and forecast results
45 lst2 <- list() #List for tsibble of train split
46 set.seed(257) #Seed for reproducibility
47 for (id in as.list(unique(data$gID))) {
48   df <- filter(data, gID == id)
49   test <- ftest(df, p)
50   ##autoARIMA
51   a_fit <- ftrain(df, p) |>
52     model(ARIMA(Demand, stepwise = FALSE, greedy = FALSE, trace = FALSE))
```

```
54   arimafc <- forecast(a_fit, h=nrow(test))
55   test$model.autoARIMA <- a_fit$`ARIMA(Demand, stepwise = FALSE, greedy = FALSE, trace =
       FALSE)`
56   test$autoARIMA <- as.numeric(arimafc$.mean)
57   ###ETS
58   e_fit <- ftrain(df, p) |>
59     model(ETS(Demand))
61   etsfc <- forecast(e_fit, h=nrow(test))
62   test$model.ETS <- e_fit$`ETS(Demand)`
63   test$ETS <- etsfc$.mean
64   ###Prophet
65   p_fit <- ftrain(df, p) |>
66     model(prophet(Demand))
68   pfc <- forecast(p_fit, h=nrow(test))
69   test$model.Prophet <- p_fit$`prophet(Demand)`
70   test$Prophet <- pfc$.mean
71   ###NNAR
72   n_fit <- ftrain(df, p) |>
73     model(NNETAR(sqrt(Demand)))
75   nnarfc <- forecast(n_fit, h=nrow(test), times=100)
76   test$model.NNAR <- n_fit$`NNETAR(sqrt(Demand))`
77   test$NNAR <- nnarfc$.mean
85   lst[[length(lst) + 1]] <- test
86   lst2[[length(lst2) + 1]] <- ftrain(df, p)
87 }
88 Df1 <- bind_rows(lst)
89 Df2 <- bind_rows(lst2)
90 write.csv2(Df1, "RForecasts_test.csv")
91 write.csv2(Df2, "RForecasts_train.csv")
```

Listing E.11. Forecasting R: Applying Models

This long snippet in Listing [E.11], is essentially the entire source code for the forecasts. Lines 29-41 defines functions for splitting the input data into *train* and *test* splits, given a predefined percentage (given on line 30). Lines 43-47 defines lists for storing the results, and the *set.seed()* arguments, sets a predefined random seed, so that functions like *NNAR* always uses the same random seed for reproducibility. Lines 47-48 creates a list of all demand groups and loops through this list.

Afterwards, a placeholder dataframe is defined and filtered by the group ID in the given iteration, allowing the 28 groups to be fed through the same forecasting pipeline iteratively.

From here, each forecasting model, *autoARIMA*, *ETS*, *Prophet*, and *NNAR* is applied iteratively to the placeholder dataframe, containing the relevant group demand.
For *autoARIMA*, the model is fit to the training data and has the arguments *stepwise* = `FALSE`, *greedy* = `FALSE` (full neighbourhood search), which forces more time and computationally intensive model selection, theoretically ensuring better (or more advanced) model selection. The selected (S)ARIMA model is then applied to forecast with a period (horizon) equal to the length of the *test*-split of the group demand dataframe.

This logic is also applied to the remaining three models:

1. Model is fit to the out-of-sample data (train)
2. Selected model predicts a period equal to the length of the remaning sample (test)
3. The results are stored in the test dataframe.

For *NNAR*, as seen on line 76, the input demand is transformed by the square root, aptly called *sqrt*(). This was suggested by Hyndman and Athanasopoulos [2023] to ensure non-negative forecasting results.
In lines 85-86, the lists from 44-45 will iteratively store the results of the forecasts. The results are then concatenated in *Df1* and *Df2*, and finally, output as csv-files for further use in Python.

```python
import re
from natsort import natsorted
import os
#Ensure Graph folder path exists
if not os.path.exists("fcastgraphs"):
    os.mkdir("fcastgraphs")
testdf = pd.read_csv("RForecasts_test.csv", sep=";", decimal=',') #df containing forecats and test set
traindf = pd.read_csv("RForecasts_train.csv", sep=";", decimal=',') #df containing training set
#Formatting columns
testdf = testdf[['Date', 'Demand', 'gID', 'model.autoARIMA', 'autoARIMA',
                 'model.ETS', 'ETS', 'model.Prophet', 'Prophet', 'model.NNAR', 'NNAR']].round(decimals=2)
precomp = re.compile(r'[><]')
for col1, col2 in zip(['autoARIMA', 'ETS', 'Prophet', 'NNAR'], ['model.autoARIMA', 'model.ETS', 'model.Prophet', 'model.NNAR']):
    testdf[col1] = np.where(testdf[col1] < 1, 0, testdf[col1])
    testdf[col1] = np.round(testdf[col1], 0)
```

```python
21     testdf[col2] = [precomp.sub('', x) for x in testdf[col2]]
22 traindf = traindf[['Date', 'Demand', 'gID']].round(decimals=2)
23 traindf['Fcast'], testdf['Fcast'] = 'Train', 'Test'
24 df = pd.concat([traindf, testdf])
```

Listing E.12. Forecasting Python: Data Preperation

Even though it is possible to calculate MASE and create plots in R, it is desirable to further manipulate the data, and it was chosen to handle visualization and further manipulation in Python due to the gap in coding proficiency between the two languages, i.e. it was 'easier' to do in Python. In Listing [E.12], lines 3-5 imports the packages the built-in `re` and `os`, where the former is applied later in line 21 for list comprehension and the latter for accessing operating system files, specifically it is used to check if a folder exists, and create it if it does not exist (lines 8-9). This folder will contain all plots for visualizing forecasting results. Lines 11-13 loads the two csv-files created in Listing [E.11], and desired columns are selected, and all values are rounded to 2 decimal places (also applied in line 23). Lines 18-21 formats the relevant 'forecasting' columns, where all values <1 are set to 0, as non-negative values are not desired. Afterwards, values are rounded to integers, as the demand unit consists of kolli's. Finally, an identifier column is added to both dataframes, thus *Train* and *Test* data can be distinguished between. Finally the dataframes are concatenated into a single dataframe.

```python
def LightGBM(df):
    from LazyProphet import LazyProphet as lp
    train = df[df['Fcast'] == 'Train'].Demand.values
    test = df[df['Fcast'] == 'Test'].Demand
    lp_model = lp.LazyProphet(seasonal_period=[7, 365.25],  # list means we use both seasonal periods
                              scale=True,
                              n_basis=10,  # weighted piecewise basis functions
                              fourier_order=10,
                              ar=list(range(1, 8)),
                              decay=.99  # the 'penalty' in penalized weighted piecewise linear basis functions
                              )
    lp_model.fit(train)
    prediction = lp_model.predict(len(test))
    prediction = np.where(prediction < 1, 0, prediction)
    return np.round(prediction, 0)

```

```
87 def MASE(df, method):
88     train = df[df[’Fcast’] == ’Train’].Demand
89     test = df[df[’Fcast’] == ’Test’]
90     prediction = test[method]
91     n = test.Demand.shape[0] #Number of periods
92     d = np.abs(np.diff(train)).sum() / (n - 1) #Sum of the n-th discrete difference divided by periods
93     errors = np.abs(test.Demand - prediction) #Absolute value of the difference between actual demand and forecast
94     return errors.mean() / d
97 def DemandClassification(df):
98     df = df.replace(0, np.NaN)
99     ADI = len(df.Demand) / len((df.Demand.dropna()))
100    CV2 = (float(df.Demand.dropna().std(ddof=0)) / float(df.Demand.dropna().mean())) ** 2
101    if ADI < 1.32 and CV2 >= 0.49:
102        dc = ’Erratic’
103    if ADI >= 1.32 and CV2 >= 0.49:
104        dc = ’Lumpy’
105    if ADI < 1.32 and CV2 < 0.49:
106        dc = ’Smooth’
107    if ADI >= 1.32 and CV2 < 0.49:
108        dc = ’Intermittent’
109    return dc
```

Listing E.13. Forecasting Python: Definitions

In the snippet in Listing [E.13], functions are defined for simplicity and repeatability for iteration. These functions include *LightGBM*, *MASE*, *DemandClassification*, and *ForecastPlot*; the latter is introduced in Listing [E.14]. The *LightGBM* functions handles training and prediction demand with the gradient boosting machine learning model, knwon as LightGBM. In this thesis, an implementation that requires minimal setup and tweaking, and is specifically made for univariate time-series is applied, dubbed *LazyProphet* by the package author [Blume, 2023].

Simply, recommended parameters are selected to train the model in lines 30-36. Specifically, since all the data is daily, the period is set to m = [7, 365.25], meaning the model will test for both seasonalities. Furthermore, the use of auto-regressive terms should be given as a range of 1 to $m + 1$ according to Blume [2023]. The rest are standard parameters, and could be tweaked to improve performance. Similarly to Listing [E.11], prediction period is set to equal length of the test demand.

The *MASE* function a python implementation of the Mean Absolute Scaled Error, introduced by Hyndman and Koehler [2006]. The *DemandClassification* function returns a string based on the interval of the calculated ADI and CV^2 values of the time-series.

```
43 def ForecastPlot(df):
44     import plotly.graph_objects as go
45     train = df[df[’Fcast’] == ’Train’]
46     test = df[df[’Fcast’] == ’Test’]
51     Plotly = [’#636EFA’, ’#EF553B’, ’#00CC96’, ’#AB63FA’, ’#FFA15A’, ’#19D3F3’, ’#FF6692’,
       ’#B6E880’, ’#FF97FF’,
52               ’#FECB52’]  # Standardized Colors
57     fig1 = go.Figure()
58     fig1.add_trace(go.Scatter(x=train.Date, y=train.Demand,
59                               mode=’lines’, name=’Train’,
60                               line=dict(color=Plotly[0])))
61     fig1.add_trace(go.Scatter(x=test.Date, y=test.Demand,
62                               mode=’lines’, name=’Test’,
63                               line=dict(color=Plotly[1])))
65     fig1.update_layout(template=’ggplot2’, title=("Time-Series of " + test[’gID’].values
     [0]),
66                        autosize=False, width=1300, height=900,
67                        font_family="CMU Serif", font=dict(size=18))
68     fig1.write_image("fcastgraphs/" + str(test[’gID’].values[0]) + "-TrainTest" + ".pdf")
72     #Prediction Graph
73     fig2 = go.Figure()
74     fig2.add_trace(go.Scatter(x=test.Date, y=test.Demand,
75                              mode=’lines’, name=’Actual’,
76                              line=dict(color=Plotly[1])))
77     for idx, method in enumerate([’autoARIMA’, ’ETS’, ’NNAR’, ’Prophet’, ’LightGBM’]):
78         fig2.add_trace(go.Scatter(x=test.Date, y=test[method], name=method,
79                                   line=dict(width=2, dash=’dot’, color=Plotly[idx+2])))
80     fig2.update_layout(template="ggplot2", title=("Predicted values " + test[’gID’].values
     [0]),
81                        autosize=False, width=1300, height=900,
82                        font_family="CMU Serif", font=dict(size=18))
83     fig2.write_image("fcastgraphs/" + str(test[’gID’].values[0]) + "-Predicted" + ".pdf")
```

Listing E.14. Forecasting Python: Definitions (2) - Plotting

The last function is presented in Listing [E.14], where the function creates and saves two plots to the folder created at the start in the code, as seen in Listing [E.12]. A color scheme is defined in lines 51-52, which ensures a desired color scheme to ensure the colors are distinguishable. In lines 57-68 the first graph is defined, which plots the *Train* and *Test* demand of the given time-series. The second plot contains the actual demand and every forecasting model, so that the resulting forecasts predictions can be visually represented. The resulting plots are locally saved, as seen on line 65 and 83.