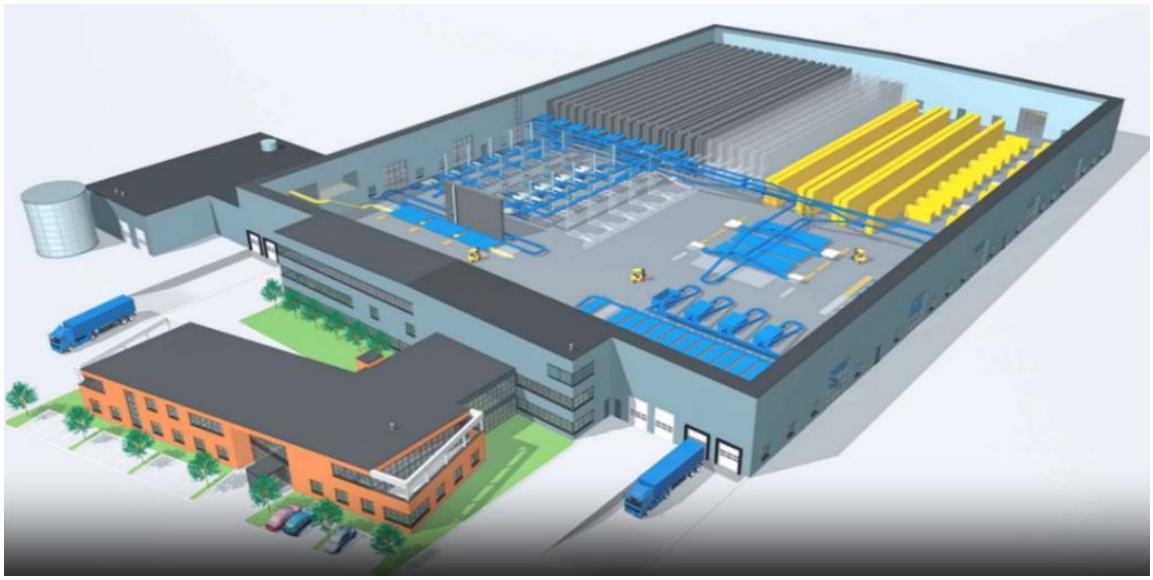
Demand forecasting at TKH Logistics

Improving alignment of staff and workload schedules at the order picking department of TKH Logistics by developing a forecasting model using time series methods.

Jurgen Joël van der Blom



Supervisors University of Twente
Dr. M. van der Heijden
Dr. Ir. L.L.M. van der Wegen

Supervisors TKH Logistics
L.J.H. Gerrits
N. Siemerink

A bachelor's Thesis
Industrial Engineering and Management
University of Twente
the 18th of June 2025

Title page

Demand forecasting at TKH Logistics

Improving alignment of staff and workload schedules at the order picking department of TKH Logistics by developing a forecasting model using time series methods.

Date
June 18, 2025

Author
J.J. van der Blom
2957787
BSc Industrial Engineering and Management
University of Twente

Supervisors
Dr. M.C. van der Heijden (UT)
Dr. Ir. L.L.M. van der Wegen (UT)
L.J.H. Gerrits (TKH Logistics)
N. Tenhagen-Siemerink (TKH Logistics)

University of Twente
Faculty of Behavioural, Management and Social sciences
PO Box 217, 7500 AE Enschede
The Netherlands
+31(0)534899111
www.utwente.nl

TKH Logistics
Elektrostraat 17, 7483 PG Haaksbergen
The Netherlands
+31 (53) 8505050
https://www.tkhlogistics.nl/

UNIVERSITY OF TWENTE. TKH LOGISTICS

Preface

Dear reader,

You are about to read my report ‘Demand forecasting at TKH Logistics.’ The research is conducted in collaboration with TKH Logistics and with the aim to graduate for the bachelor's degree Industrial Engineering and management at the University of Twente.

I would like to thank my company supervisors Berthold Gerrits and Niki Siemerink for their support and guidance at TKH Logistics, and my university supervisors Mathieu van der Heijden and Leo van der Wegen for their valuable feedback. I thank my student buddy Jurrian van Dalen for the weekly input and feedback regarding my research progress. Finally, I would like to thank family and friends for their support.

Hopefully, you will enjoy reading this thesis.

Jurgen van der Blom

Enschede, June 18, 2025

Management summary

Introduction

This research is done on the behalf of TKH Logistics which offers logistics services in a semi-automated warehouse in Haaksbergen. The research focuses on the order picking division, which is responsible for picking and packing products ordered from the warehouse. This division suffers from a mismatch between daily workload and capacity. The mismatch is predominantly caused by the staff and workload schedules, which are inadequately aligned. The main reason for the scheduling issue is the lack of trust in, and reliability of, the current demand forecasting method employed at TKHL. The research set out to provide a reliable demand forecast and an implementation plan that ensures the proposed method is trusted and used in practice. This should ultimately lead to the realisation of underlying organisational vision that a good operations environment can only be achieved when there is a level of control and calmness around the scheduled capacity and workload.

Method

Within the data warehouse of TKHL slightly more than 3 years of historic demand in terms of order lines can be obtained, whereas other data that could be used to directly predict the required staff and future workload is unavailable due to orders that are processed for different dates without following fixed rules that could be used to retrace the workload and staff per day. The best alternative to directly predicting the workload and required staff is to build a demand forecast in terms of order lines, which can serve as the basis for the daily staff and workload schedules that are made one week in advance. As processing times and the available time for processing differ for the 16 customers we forecast on customer level with 1-day time buckets. To gain insight in demand beyond the scheduling horizon we predict the future 20 days. The construction of a demand data set in terms of order lines on desired customer level provides the main model input. Analysis of the demand provided the insight that seasonal patterns within the week and over the year, as well as holiday effects, are present and should be considered. The current method, which was reconstructed in this report, struggles with capturing these patterns resulting in large prediction errors.

The research builds upon the assumption that customer demand can be predicted using historic demand as the main input and therefore explores the integration of time series methods. We try both a statistical and deep learning-based method using the NeuralProphet framework developed by Facebook in 2021. We incorporate trend, seasonality, holiday, and auto-regression effects as well as four-week order overviews that some customers provide. To evaluate the method results we compare the proposed model results with a reconstruction of the current method using time series cross validation and performance measures on overall and customer level. These measures include the MWAPE, bias, and MAD.

Results

A basic Prophet model that only incorporates trend and seasonality effects underperforms the current forecasting method. We identify three flaws; large customer bias, poor performance around holidays, and unutilized information as the order overviews that provide insightful information are not incorporated in the model. In the following model extensions, we deal with these issues by including holiday effects, autoregression effects, and order overviews. The final model that is constructed with trend, seasonality, holiday, and auto-regression effects as well as four-week order overviews reduces the MWAPE of the current method by 64.8%, the bias by 92.9%, and the MAD by 27.4% on a 7-day horizon in the test set. On a 20-day horizon the results were 57.0%, 87.8%, and 21.6%, respectively. On customer level the final model performed similar to the reconstruction for the customers that provide order overviews, as both methods directly copy the estimates from the overviews when these are available, and better for the others based on the MAD and bias. The set

norm for the model, which we deem slightly less insightful in actual model performance, was no bias and a MAPE of 10% on aggregated daily level over a 7-day horizon. We were able to attain close to no bias and a MAPE of 13.8% on the test set, using the final model. Attained performance is slightly worse than desired but considering the achieved MAPE of the current method in the test set and the relatively small gap between realised and desired performance we conclude that the performance of the proposed professional model is acceptable.

To ensure that the configured model can and will be used in practice we consider its implementation using three implementation aspects. By providing the model structure TKHL can realise the technical model implementation. Through change management literature and the proposition and description of three tailored types of vision dialog sessions we aim to provide required guidance to train and familiarize staff with the method and underlying organisational vision. Supply chain integration is discussed to further improve the model performance in the future.

Conclusions and recommendations

We conclude that the provided forecasting model provides satisfactory results over a 7-day horizon, and that up to 17 days the results are stable. Predictions further in the future suffer from the lack of available four-week demand overviews and show worse performance. To implement the model and actually solve the action problem three components should be considered; the technical implementation, change management, and the supply chain integration. We recommend TKHL to:

1. Deploy the proposed method and consider and manage the three implementation aspects discussed in this report. As for the supply chain integration and four-week order overviews be sure to also:
    a. Resolve the structural 9 day ahead prediction error in the four-week order overviews from two customers.
    b. Explore whether the bundled order overview from two customers can be provided separately.
2. Explore whether model performance can be improved through identification of more explanatory variables, further model tuning, and aggregation of small customers if possible.
3. Re-evaluate the best method for including order overviews when multiple years of data with order overviews are collected, as this may change the outcome.
4. Control model performance with a KPI, we suggest the MWAPE and distinguish between a scheduling horizon (7 days) and total model horizon (20 days) when setting targets for the KPI as they have different performance requirements.
5. Train the staff so they understand the model predictions and that they serve as source of information for scheduling decisions but cannot be blindly trusted.
6. Re-evaluate the way pick times are collected, as the current method does not reflect reality even though accurate estimates will be beneficial for scheduling using the demand forecast.
7. Establish the internal processing rules for the workload schedules and/or conduct further research towards optimal workload and staff scheduling procedures using the predictions from the method provided in this research.
8. Follow this research procedure to build a similar model for the incoming product flow to support decision-making for inbound staff and workload schedules.

Contents

Glossary

TERM	DEFINITION
DPS STORAGE	Dry Pack Storage. Storage of light products that are too long to fit in the MLS boxes and on pallets.
ISOWEEK	The ISO calendar is a leap week calendar system. It uses 7 weekday cycles, like the Gregorian calendar. Weeks start with Monday. An ISO year has 52 or 53 full weeks (364 or 371 days) and starts on the week with the first Thursday in the new year.
MLS STORAGE	Mini Load Storage. Storage of products lighter than 20kg and within the following dimensions 667x366x297 mm (L/W/H) such that it fits in boxes that can be stored in the automated box storage.
MPS STORAGE	Manual Pallet Storage. Storage of products on pallets. Used when MLS and DPS are deemed unfit.
MPSM	Managerial Problem-Solving Method. A method proposed by H. Heerkens to solve managerial problems in a systematic way by following the seven phases in the iterative cycle.
NET WORKING HOURS	The cumulative net number of hours staff works at the order picking stations excluding sick leave, lunch time etc. (e.g. a full-time order picker works 8 net hours per day)
NORMAL PICKING (NP)	Subdivision of order picking that only picks products from the MLS storage.
ORDER LINE	An order line is a specific line in an order that represents a single item. There can be multiple order lines in an order, each corresponding to a separate line item that contains a unique combination of products, quantities, and other relevant details.
ORDER PICKING	Selecting and retrieving items or products from inventory to fulfil customer orders and placing them in boxes according to the order statement.
ORDER STATEMENT	A statement which the client provides to TKH Logistics with information about the amount of order lines they (expect) to order in the (near) future.
SPECIAL PICKING (SP)	Subdivision of order picking that can also pick products stored in the DPS storage, next to the MLS storage.
WEM	WEM is a no-code platform that enables businesses to build custom web applications and automate workflows with enterprise-grade capabilities, all without writing code. The abbreviation stands for “Web Enterprise Modeler.”

1. Introduction

In this chapter, we will construct the research design. For this we will introduce TKH Logistics in S1.1, define the research motivation in S1.2, identify the problem and set the research goal in S1.3, discuss the research scope in S1.4, and discuss the plan of approach including the research design in S1.5.

1.1 Company description

TKH Logistics is a subsidiary of the stock listed TKH Group. With just over 100 employees, of which approximately 60 fte, TKH Logistics offers logistics services in their semi-automated warehouse in Haaksbergen. The services consist of, but are not limited to, storage, bulk breaking, drop shipments, and cross-docking. Within the warehouse there are several departments of which the three key units are inbound, order picking and outbound. Simply put, the inbound station manages the arrivals by checking, documenting, and storing the incoming products. When a product stored in the warehouse is ordered by one of the clients of TKHL, the product is collected and packed by the order picking division and sent to the outbound division to be prepared for transport. A more detailed visualization of the in- and outbound process is provided in Appendix *A.1 Warehouse process description*. Following this procedure TKH Logistics processes product orders for seven subsidiaries within the warehouse visualized in Figure 1. In 2024, approximately 1,000,000 outbound order lines were processed in the warehouse. These order lines consist of a wide range of various products but are mostly technological consumer products such as cables and electronics or industrial machine components such as bolts, threads, and engine parts. Each subsidiary has its own variety of products such that the type of products per client are relatively similar, but very different to that of the other clients. Client A may have many cables, whereas client B handles large machine components.

The research focuses on the order picking division, which is responsible for picking and packing the ordered products. This division consists of two subdivisions, namely normal picking (NP) and special picking (SP). At special picking, products from both the automated box (MLS) and pallet (DPS) storage can be picked, whereas normal picking can only be used to pick products from MLS. These two divisions retrieve, and pack ordered products located in the boxes from the semi-automated storage (MLS) and pallet racks (MPS). A small fraction of the products is stored in a dry pack storage (DPS) specifically designed for light products that are too long to fit in the MLS boxes. Ordered products that are stored at the dry pack storage (DPS), which was 1.46% of the total number of order lines in 2024, are picked by staff from quality control (QC); the division that checks a fraction of the packed products to ensure the orders are packed correctly. Like the inbound and outbound divisions, the order picking subdivisions consist of several stations at which staff can work.

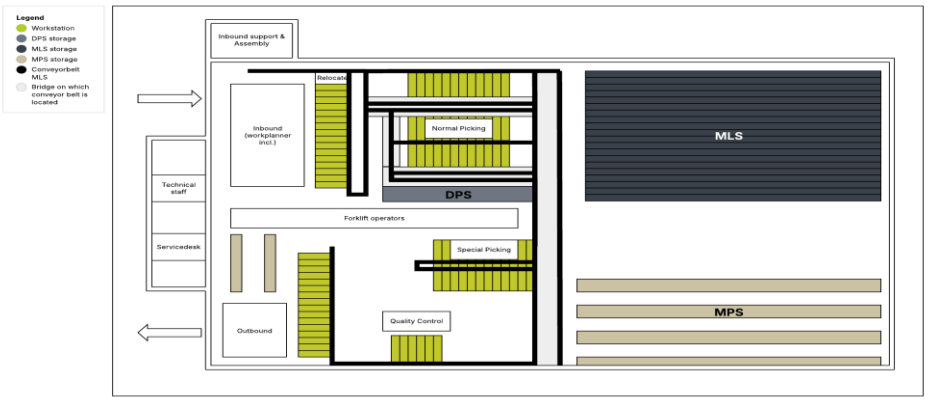


Figure 1: TKH Logistics Warehouse

1.2 Research motivation

In order to be able to process all the orders in time, whilst keeping staffing costs down, order picking staff schedules have to match the daily workload. Matching the number of net working hours with the workload is therefore of the utmost importance to utilize process efficiency. Currently TKH Logistics struggles with achieving this for their two order picking subdivisions. This leads to high fluctuations in work pressure, overtime hours, and unproductivity.

To illustrate this fluctuation in work pressure, the productivity for 2024 of the normal and special-order picking subdivisions was analysed. If staff hours and demand are balanced, we would expect that the productivity per staff hour would randomly fluctuate around a constant regardless of the total number of order lines on a day, as the number of staff hours would be in line with the total workload. In reality however, staff is likely processing more order lines per hour on a day with many order lines compared to days with lower volumes as shown in Figure 2. The productivity in terms of processed order lines per staff hour is higher on a day with more order lines on average. This could be the result of frequent overcapacity that is only used on busy days, undercapacity that is dealt with by staff working faster and less precise (resulting in mistakes) on busy days, or some other reason. Either way, we can conclude from this result that the workload per staff hour is currently not constant. In other words, the number of staff hours is not aligned with the total number of order lines on a day.

When statistically testing the linear relationship between the two variables we find we should reject the null hypothesis that the productivity and the total daily number of order lines are not correlated for both SP and NP even for low alpha when using the Pearson correlation coefficient (details in Appendix *A.2 Pearson correlation test*). Additionally, the unbalance seems to be larger for special picking than normal picking as the slopes indicates the productivity of SP increases by 1 order line per staff hour for every 54 (1/0.0186) additional order lines, where this is 152 (1/0.0066) for NP.

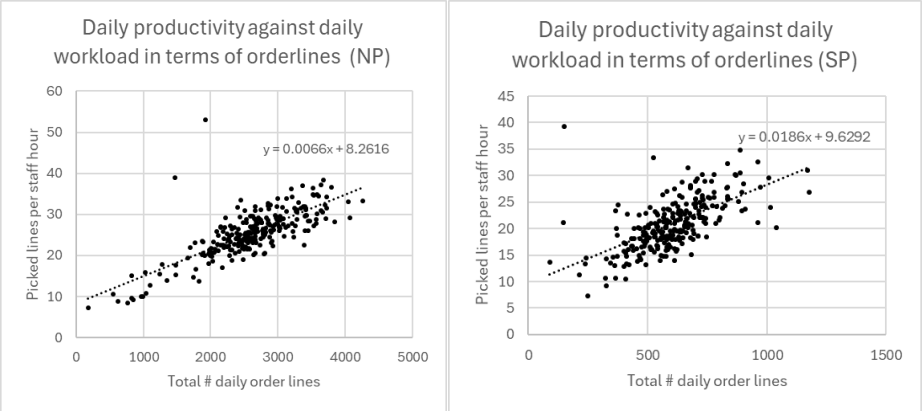


Figure 2A & 2B: NP and SP picked order lines per net staff hours against total # order lines per day in 2024

To limit the mismatch, the team management tries to relocate the workforce between the different (sub)divisions the moment a mismatch is observed. This requires a lot of flexibility from employees and is often not effective because the workload across the divisions is positively correlated such that all divisions have either too many or little staff generally. For that reason, TKH Logistics wants to explore the possibilities for effectively reducing this imbalance between workload and staff hours.

1.3 Problem identification

In this section, the core problem is selected by following the four steps proposed by Heerkens & Van Winden. The action problem is identified in S1.3.1. A problem cluster is constructed in S1.3.2. The core problem to tackle is selected in S1.3.3 and quantified in S1.3.4. In S1.3.5 the main research question is formulated and in S1.3.6 we define the research goal and deliverables.

1.3.1 Identification of action problem

TKH Logistics observes the results of a mismatch between the order processing capacity at the order picking division and the daily workload in the form of process inefficiencies and inconveniences, such as overtime. Through observation and interviews with the directors, team management, and order picking staff a “quick-and-dirty” problem overview (Appendix A.3 Quick and dirty problem overview) was constructed from which the following action problem appeared: There is a mismatch between daily workload and processing capacity at the two order picking subdivisions.

1.3.2 Problem cluster

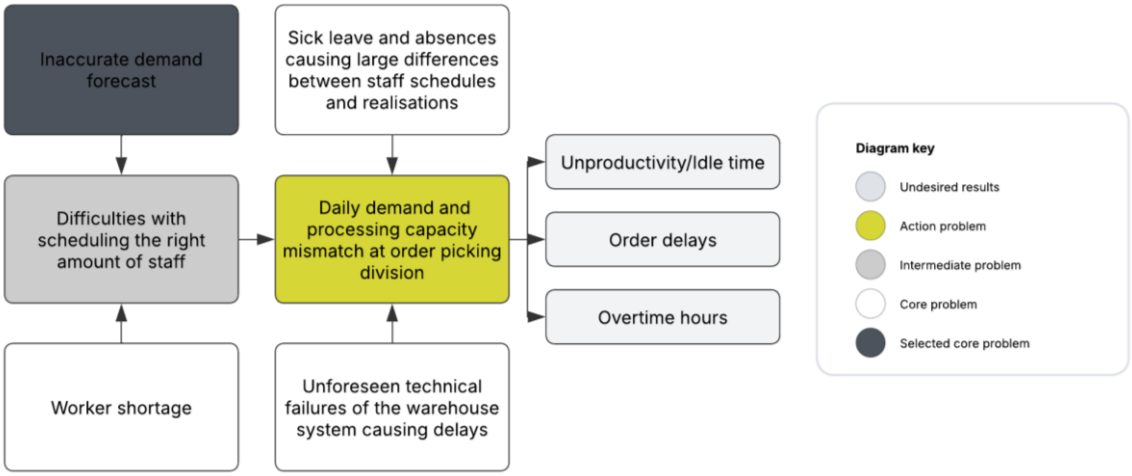


Figure 3: problem cluster

With the action problem as the starting point the Why-Why Analysis approach was used to identify the core problem(s) as it is “an easy-to-use approach for arriving at a root cause” (Ramu, 2022). We construct a problem cluster to find the core problem. On the surface the mismatch between daily workload and the actual capacity of the order picking division, which mostly depends on the amount of scheduled labour, is observed in the form of unproductivity, relocation of staff, overtime, and missing client deadlines.

The mismatch between daily demand and capacity is caused by the staff scheduling procedure for the order picking division, which provides staff schedules that do not match the daily workload. Technical failures of the warehouse system, which are caused by the worn-out and outdated conveyor and WMS system, and a high sick leave / absence rate are the other two directs causes of the mismatch.

The mismatch between the daily workload and capacity is caused by a worker shortage that makes it hard to schedule the number of staff that is required to manage the workload. Another reason is the inaccuracy of the demand forecast that is used as input for the staff schedules.

1.3.3 Motivation of core problem

From the identified four core problems three are already being worked on by TKH Logistics. The implementation of a new warehouse management system (WMS) is nearby which should deal with the software failures. Team management is working on plans for reducing the sick leave rate to improve the alignment of original staff schedules with their realisation and is hiring additional staff. This leaves one main core problem that should be tackled before the problem can be solved.

The scheduler struggles to estimate how much staff he should schedule. Using the demand forecast, the required number of staff can be estimated. However, the demand forecast is currently deemed too unreliable. Instead, schedules are now made based on experience, which also does not work well. TKH Logistics requires a demand forecast that the staff schedulers trust and can base their scheduling decisions on. This is the chosen core problem. Tackling the problem will allow scheduling staff and workload more effectively by utilizing the information provided by a demand forecast.

1.3.4 Measurement of norm and reality

With the core problem, we can formulate a problem statement to clearly define the norm, reality, and gap. The ultimate goal is to match daily workload and staff schedules and for that we require an accurate and trusted demand forecast, which is not available now. The forecast error is defined as "the difference between the actual value and what was forecasted" which can be formulated as $e_i = X_i - f_i$ where the error (e_i) is denoted as the realisation (X_i) minus the forecasted volume (f_i). Common measures of error are the bias, mean absolute deviation (MAD) and the mean average percentage error (MAPE). MAPE is the most widely used measure for calculating the forecast error and is preferred to the MAD as an accuracy measure because of its lack of dimension, which makes it "nice for communication purposes" (Charles W, 1995).

Using the MAPE the forecast inaccuracy of total number of order lines per day with the current method, in which Demand for the current week is predicted every Monday, was 30.8% over the first eight weeks of 2025 *(Appendix A.4 Current forecasting performance).* Part of this error seems to be the result of a systematic forecasting bias in which the demand is overestimated, as the data showed a bias of -3.5%. The request of TKH Logistics is to aim for an aggregated forecast inaccuracy (MAPE) of 10% on a daily base as they are convinced this provides sufficient accuracy to build up trust and use it to drive the staff and workload schedules. The remaining inaccuracy can be handled by revising the workload schedules using the freedom for the processing day that some subsidiaries provide.

Using historic demand of 2023 and 2024 yearly trend, seasonality (daily and monthly), and level were decomposed to determine the systematic and random component of aggregated customer demand by dividing the remaining random component by the actual demand, just like when calculating the MAPE. The random component was 33.10% of total demand variability (*A.5 Decomposition of demand 2023-2024*). "On average, a good forecasting method has an error whose size is comparable to the random component of demand (Chopra, 2019)." As this method did not yet capture potentially more complex patterns and include order overviews provided by some customers, reducing the random component to a MAPE of 10% seems like a realistic goal. As "Long-term forecasts are usually less accurate than short-term forecasts" (Chopra, 2019) this accuracy should be achieved for at least the period for which the staff schedules are made, which is 7 days ahead.

Problem statement

TKH Logistics aims to remove the bias and decrease the MAPE of their daily demand forecast, in order lines, from 30.8% to 10% on a 7-day horizon.

1.3.5 Research question

Reformulating the problem statement provides the research question for this research. We include that we will do so through the use of time series methods, as demand predictions are generally made using forecasting theory. More precisely, we use time-series methods, as these are most appropriate when future demand is related to historical demand, growth, and seasonal patterns (Chopra, 2019).

"How can TKH Logistics remove the bias and decrease the MAPE of their daily demand forecast, in order lines, from 30.8% to 10% on a 7-day horizon using time series methods?"

1.3.6 Research goal and deliverables

The research goal is to predict the daily amount of outbound order lines per day with a mean average percentage error of at most 10% and provide guidance in the implementation of the method. The deliverable is an operational demand forecast of outbound order lines and implementation plan.

The demand forecast may be aggregated to the total number of order lines for the order picking divisions and does not have to separate between order lines picked by normal, special picking or DPS as the ratio of order lines for normal, special, and DPS is mostly constant as shown in Figure 4: fraction of order lines per subdivision 2024. On average approximately 79.95% of the order lines was picked by normal, 18.59% by special and 1.46% by DPS in 2024. Next to that, all staff located at special picking can also work at normal picking and some employees at normal picking can also work at special picking, such that the employees are mostly exchangeable. For DPS only one employee is required, and this employee can also work on the other subdivisions. Therefore, it can be aggregated.

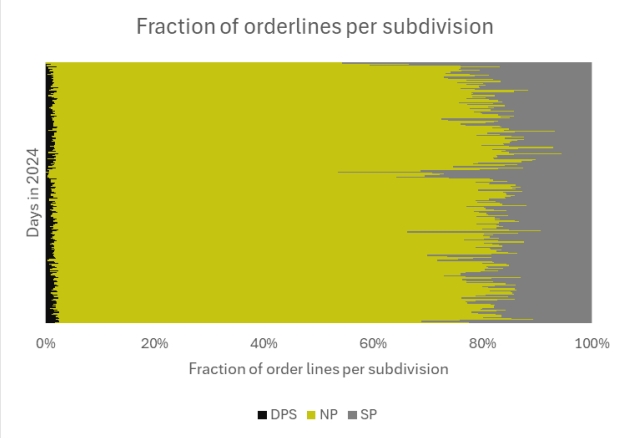


Figure 4: fraction of order lines per subdivision 2024

The forecast should be disaggregated to client level for two reasons. First, the processing time depends heavily on the client as shown in Table 1. These estimates were made by analysing the available pick times data per customer. As customers N, O, and P order the same product variety its estimate was bundled. This data consists of a "pickstart" and "pickend" time for each order and is measured from the moment the first pick is finished until the moment the last pick is finished. This implies the first pick is not recorded. Therefore, the estimate is made using orders of two or more picks only and for each order the average time per pick was once added to account for the missing pick. Next to that, because of this method, the time in between two orders is not recorded. So, the estimate is not directly convertible to total number of required hours due to this time gap in between shipments. Still, it shows that the pick times significantly differ per customer. This would not be an issue had the ratio of orders from the customers been constant. However, this ratio fluctuates. To know how much staff to schedule based on the demand forecast, the scheduler should have insight in which customer will be ordering. We must distinguish between the customers.

CUSTOMER	A	B	C	D	E	F	G	H	I	J	K	L	M	N,O,P
AVG. TIME PER ORDERLINE (MIN)	1.8	2.1	4.0	7.8	4.8	14.2	1.5	1.6	3.2	1.9	1.6	3.3	3.1	4.1

Table 1: 2024 Estimated order line pick time per client

Second, one customer places some of its orders a few days before the order has to be sent. For example, Company X has an order each Monday for which it provides at least 50% of the order information the Monday before, and the other half on Wednesday. So, although this order is due Monday, all of it can already be picked in the week before. TKH Logistics wants to use the additional time available for processing these orders to balance the workload over the days evenly according to their internal processing rules. These include rules such as picking the products for a given company in the week before the products have to be shipped. A comprehensive overview of the rules can be found in *A.6 Internal processing rules*. To allow the desired workload reallocation, it is necessary to distinguish between customer demand.

In short, the demand forecast should be disaggregated to customer level and predict demand for each departure date. That way, the scheduler can use the information to create a workload and staff schedule. The forecast will be used on an operational level, for daily staff and workload scheduling. Therefore, we use daily time buckets. Despite schedules being made one week in advance, total horizon is chosen to be four weeks to give greater insight in future demand on an operational level, by request of TKHL.

1.4 Scope and limitations

Considering the scope and limitations of the research ensures that expectations and resources are managed, and that the research subject is well-defined. The research aims to contribute to creating an operation environment with a level of control and calmness around the staff and workload schedules and their alignment and will attempt doing so by introducing a better performing time series method than the current one, and a plan for its implementation as input for the staff and workload schedules.

Forecasting demand can be done for both the in- and outgoing flow at TKHL. However, this research focuses on forecasting the outbound order volume as the action problem concerns the order picking division, which manages outgoing products only. Ideally, the inbound division would also be included in the research as their staff requirements also depend on product volumes and perceives the same problems as the picking division. To manage time, the decision was however made to focus strictly on the outgoing flow and the picking division.

The ideal workload and staff scheduling procedure for demand are excluded from this research. The research focuses on providing reliable demand predictions and an implementation plan that provides guidance for implementing the provided method to steer the workload and staff schedules. Improving the workload and staff scheduling procedures may be the topic of a follow-up research project and is excluded from this research.

External data that might be correlated with demand, such as macro-economic trends, are disregarded in this research as its impact on demand is assumed to be relatively insignificant compared to the additional time that would be required to evaluate and include these sources. The research only focuses on historic demand, order overviews provided by customers, a set of national holidays and scrap orders for one customer of which the effect was identified through outliers in its demand.

1.5 Plan of approach

The plan of approach, phase two of the Managerial Problem-Solving Method (MPSM), provides the outline for tackling the research problem defined in S1.3. A theoretical framework of the forecasting method will be developed in S1.5.1. In S1.5.2 we construct the research design in which phases 3 to 6 of the MPSM are incorporated.

1.5.1 Constructs

In this research the researcher obtained access to the customer demand of TKH Logistics from the beginning of 2022 up to and including week 13 in 2025. Next to this quantitative data source, one large customer provides a four-week overview of the daily amount of order lines it expects to place for five of its eight divisions every day. From the 28th of October 2024 these daily overviews are collected in a database. Evaluating their accuracy over the period from October 28th, 2024, until April 1st, 2025, shows that these overviews are, especially when further away from its realization, quite inaccurate as shown in Figure 5 by the symmetric mean average percentage error (SMAPE).

The accuracy is shown in terms of SMAPE instead of MAPE as some of these divisions have instances where daily demand is 0 resulting in issues with its calculation. The symmetric MAPE (1) handles these issues by dividing the absolute error by the average of the forecasted and actual demand:

$$\frac{1}{n} * \sum_{i=1}^{n} \frac{|f_i - X_i|}{(|X_i| + |f_i|)/2} \qquad \textit{where } X_i \textit{ is the realisation, and } f_i \textit{ the forecasted volume}. \qquad (1)$$

Simply using the provided overviews for these customers might therefore be suboptimal as we might be able to obtain better predictions by utilizing the information gained from historic data for them as well. Especially when we consider that staff schedules are made on Thursday for the week ahead. Upcoming Friday is then six days ahead, for which the SMAPE of all provided overviews was over 20%. As these are significant inaccuracies, we will experiment with whether including these overviews as input in a forecasting model for these divisions provides better results than the overviews themselves.

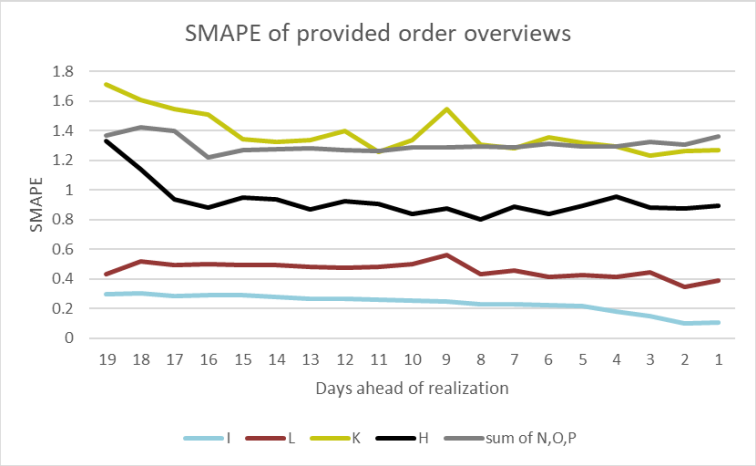


Figure 5: SMAPE of provided four-week demand overviews per customer

So, to predict future demand two quantitative data sources were gathered from TKHL:
1. Historic demand in terms of daily number of order lines per client, covering the period from January 2022 up to and including March 2025.
2. Four-week demand overviews (provided by one of the customers for five of its divisions) from October 28, 2024, up to and including March 2025.

The research will consist of finding out how historic order volumes can be used to predict future demand using time series methods as shown in grey in Figure 6. In addition to that, we will explore the presence of sporadic demand during and around holidays. For the customers providing a four-week order overview, we will evaluate whether using the overviews as exogeneous variables or direct input can improve model performance, as shown in green in Figure 6. As the order overviews are collected since the end of October 2024, this data is available over a period of five months only. Imputing 2.5 years using data from a period of 5 months is unlikely to yield accurate results, so we will consider using the overviews as explanatory variables as an alternative method in which national holidays and yearly seasonality will not be considered as we lack data in that case.

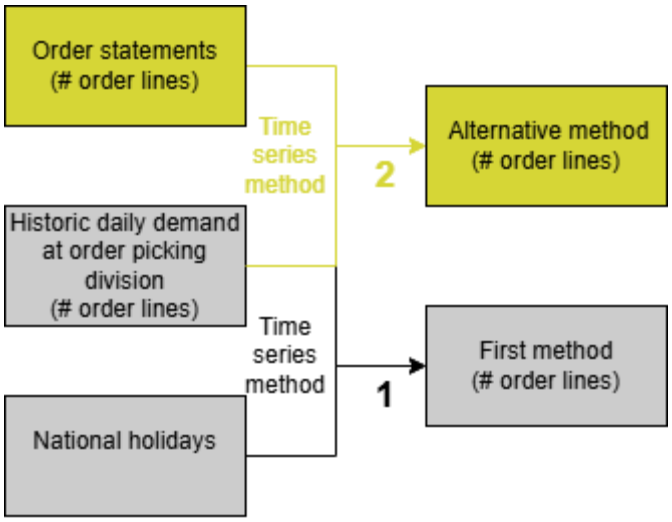


Figure 6: Theoretical framework

1.5.2 Research design

In this section we will construct the research design according to phases 3 to 6 of the MPSM, which was chosen as the global problem-solving approach because of its wide applicability in various situations (Heerkens et al., 2021). Phase 1, the problem identification, was already conduced in S1.3 and this section encompasses phase 2, the plan of approach. Phase 7, the evaluation phase, is not included in this research as it requires the implementation to be conducted, which will not be done in this project but can later be done by TKHL.

Although the MPSM provides a global research structure, it does not manage the particular tasks involved in a machine learning project. As training the time series method will be one of the main activities in this research, we introduce the machine learning pipeline (Subasi, 2020) to handle the activities that are required to train the forecasting model. The proposed steps include the collection and preprocessing of data, model selection, training and evaluation, and finally the implementation. We assign the activities to MPSM phases 3 to 6 in the research design like shown in Figure 7.

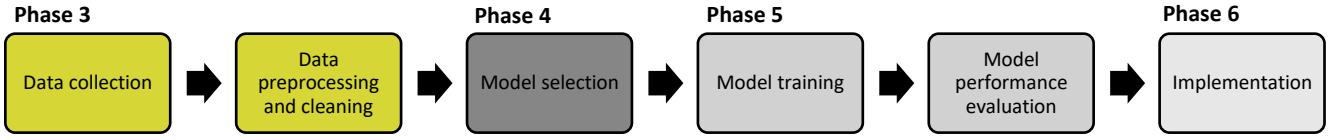


Figure 7: Machine learning approach integration within MPSM phases 3 to 6

Phase 3 – Problem context analysis - RQ1: How does TKH Logistics forecast demand, and what is the current forecasting quality?

To answer the main research question, we shall re-examine the current situation in greater depth, which we will do in Chapter 2. We produce a preprocessed demand dataset on the desired aggregation level, as required by Figure 7, and describe the current forecasting method. Using the dataset and procedure we will reconstruct the current method on the desired aggregation level to serve as a benchmark for the deliverable. Finally, we will assess the presence of seasonal patterns within the week and year as well as potential holiday effects to decide whether the deliverable should account for seasonality.

1.1 How does TKH Logistics currently predict future demand?
1.2 What is the forecasting quality of the current method on desired aggregation level?
1.3 Are customers' demand influenced by seasonal variations, and if so, in what ways?

Phase 4 – solution generation - RQ2: What time series methods that allow integration of exogeneous variables can predict future demand accurately?

In Chapter 3, the identification of time series methods that allow exogeneous variables will take place to determine which methods may be used to forecast the future number of order lines using the historic demand, seasonality if assumed present, and provided order statements if provided by the customer. Literature shall be studied to familiarize how identified models can be setup, trained, evaluated, and compared. As for studying possible methods for model evaluation, it shall be considered how to evaluate the individual models (per customer). This should include methods that can be used when demand is sometimes zero, such that standard methods like MAPE can no longer be used. The customer models should in the end be evaluated altogether on a daily basis as the models will be used alongside each other to predict the total demand, so a method for that purpose shall be identified as well.

2.1 What time series methods allowing exogeneous variables can effectively predict the demand?
2.2 How can the method performance be validated and compared using validation techniques and performance metrics?

Phase 5 – solution selection and validation - RQ3: How can demand be predicted using historic data and order statements?

In Chapter 4 a method will be selected, and the model (extensions) will be trained and evaluated. To structure the model training KDD, SEMMA and Crisp-DM, three industry-standard methods in the field of data science and mining, were compared. SEMMA, which consists of sampling, exploring, modifying, modelling and assessing the accuracy in a cycle of continuous improvement, was chosen as the method to follow in the training process because of its narrower focus compared to the other two methods (Azevedo, 2008). This is desired as the broad structure is already provided by the MPSM and approach visualized in Figure 7. The model(s) will be evaluated using the validation techniques and metrics to be identified in phase 4, to ensure model validity as required when conducting a forecasting project (Chopra, 2019). This performance analysis will also include the comparison of the results with and without integration of provided customer order overviews with the benchmark constructed phase 3. It will be determined whether the proposed methods improve prediction results and whether the inclusion of the four-week overviews is desired. Finally, the impact of the proposed intervention may be illustrated by comparison of the results with the benchmark.

3.1 Which identified time series method fits the objective best?
3.2 How do the model (extensions) perform compared to the benchmark?
3.3 Does including the order overviews from customers improve prediction results?

Phase 6 – Implementation plan - RQ4: How can the forecasting method be implemented at TKH Logistics?

To implement the solution in practice, the forecast needs to be implemented. In Chapter 5 we will write an implementation plan for TKH Logistics to illustrate how the constructed models may be used in practice to provide insight in future demand and serve as input for the staff and workload schedules.

2. Context analysis

In this chapter, we will acquire and analyse available demand data in S2.1 and S2.2 as we need this data to answer RQ1.2. In S2.3 we answer RQ1.1, regarding the current forecasting procedure at TKH Logistics. In S2.4 we build a reconstruction of the current forecasting method to answer RQ1.2, regarding the current forecasting method performance, in S2.5. For the reconstructed model we use the information and data acquired in the first three sections of this chapter. In S2.6 we evaluate the presence of seasonal patterns and in S2.7 we decompose demand using identified seasonal patterns to answer RQ1.3. In S2.8 we present the findings from this context analysis.

2.1 Data preprocessing

Data collected by the warehouse management system (WMS) and stored in the data warehouse of TKHL was obtained and pre-processed to acquire a dataset with historic demand per day in terms of order lines per customer by merging a shipment and outbound orders dataset using the "shipment ID" as link. The shipment dataset contains the day of departure (potential pick date(s) can be derived from it according to *A.6 Internal processing rules*) and name of the customer, whereas the outbound orders dataset contains the number of order lines per shipment. We need both for the desired dataset. Before being able to merge the two datasets the data had to be cleaned. In the shipments data, the desired level of customer disaggregation was not met, requiring the analysis of other data, such as shipment classes, to identify all the unique customers.

Documented departure dates are not always correct. In particular, certain shipments had departure dates on Saturdays or Sundays, which is impossible as TKHL is closed during the weekends. This could be the result of the outbound division manually overwriting the true date of departure due to a letting restriction in which it is sometimes impossible to place all the departures on the true date. In that case they overwrite the true date and place it on a Saturday or Sunday. In this case, the original date will be correct. We replace the departure date by the original date as this is the true departure date in that case. If the original date is also in the weekend the shipment is an online order, which can be placed during the weekends but will be processed on Mondays. So, these departure dates are set to the Monday after the weekend. Six inconsistencies with date format were found in the dataset, which were manually corrected. Preprocessing the outbound order dataset required finding the number of order lines per shipment, as this dataset contains a row for each order line number within the shipment. This was done by taking the maximum order line value for each shipment ID as the order lines for each shipment count upwards from 1 such that the highest value is also the total number of order lines. The preprocessing steps can be found in *A.7 Data collection, cleaning, and transformation of customer demand dataset* and are summarized in Figure 8. The dataset with dates from 2022 up to week 13 2025 contains the number of order lines per day and customer.

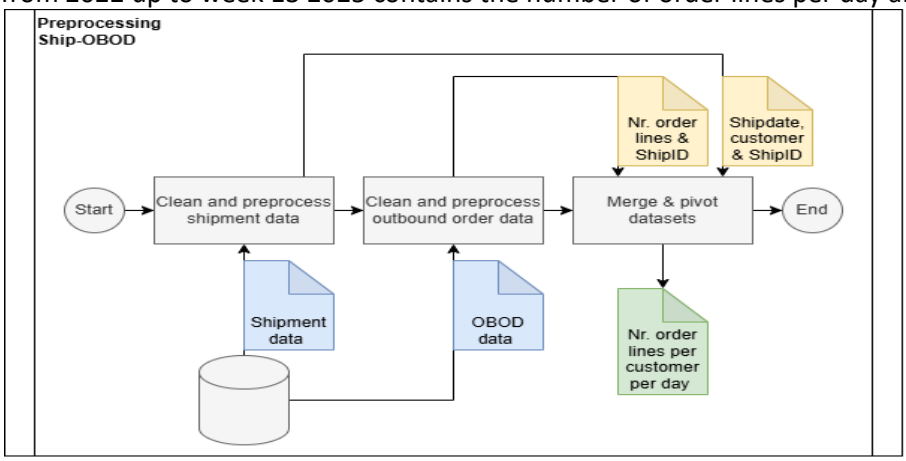


Figure 8: Preprocessing shipment-outbound datasets

2.2 Historic demand

The acquired dataset in previous section allows disaggregating the total daily demand in order lines as shown in Figure 10 to customer level. The contribution to the total number of order lines depends heavily on the customer, ranging from less than 0.05% to 32.4% in the period from 2024 as shown in table 2.

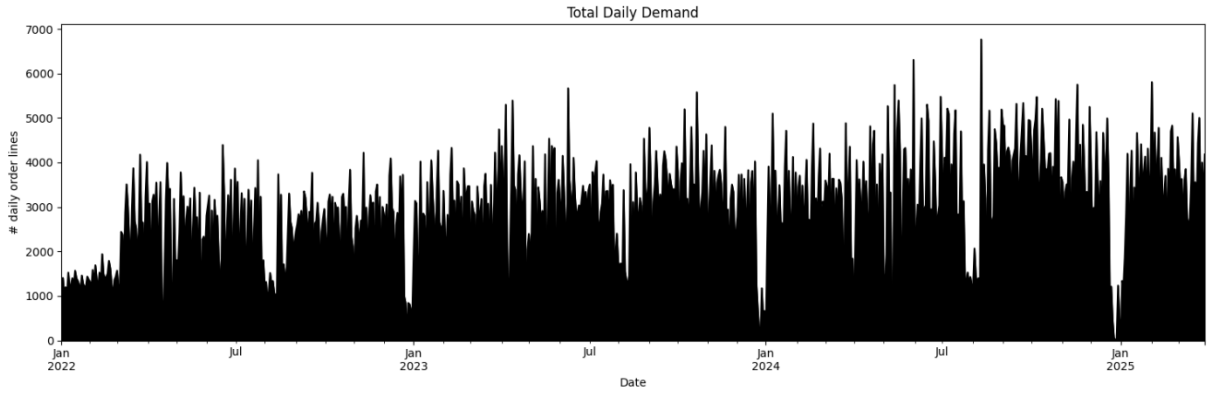


Figure 9: Total daily demand TKHL

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
0.04	0.07	30.85	1.13	0.65	2.01	0.02	0.45	12.18	1.23	11.43	32.24	4.29	0.0	3.16	0.26

Table 2: Contribution to total order lines per customer A - P (%) over 2024 up to week 13 2025

When we disaggregate the demand to customer level we observe 16 unique demand patterns, shown below in Figure 10. From the potential outliers in the customer demand, only the two outliers of customer F can be explained. These were pre-announced scrap orders, in which old inventory was sent away. Some customers are closed during the construction holiday and Christmas, explaining overall reduction in demand during these periods in Figure 10. The composed data set contains historic demand on desired aggregation level and provides the input for the training and evaluation of time series methods in the upcoming chapters. Using this dataset we can proceed the research.

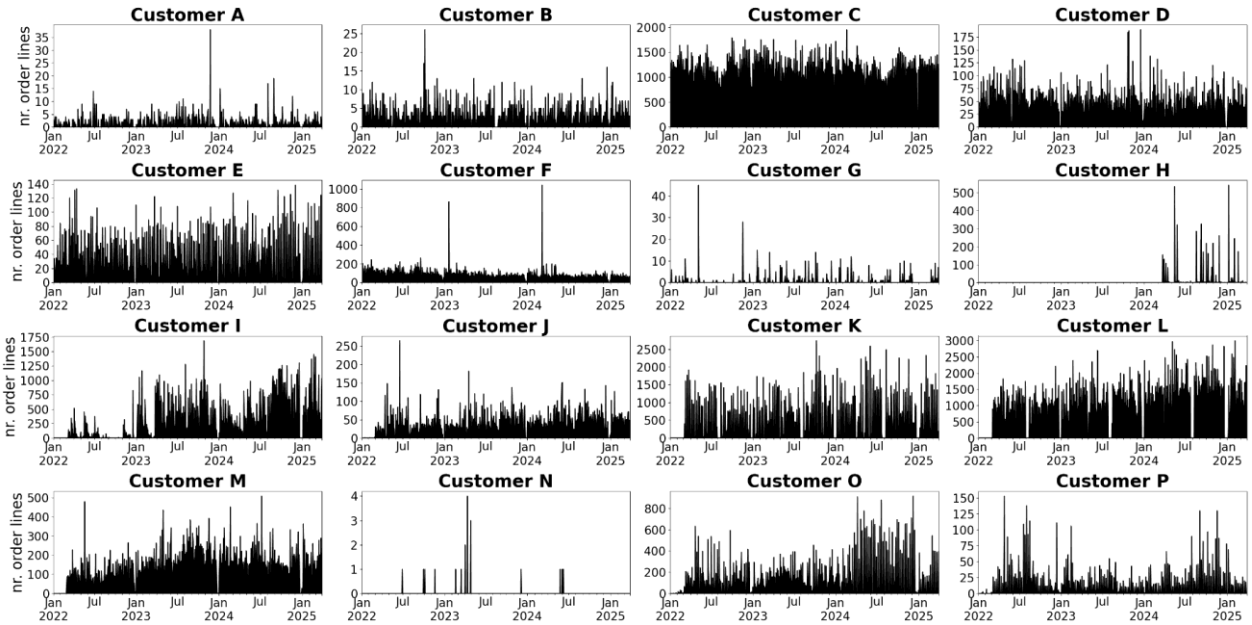


Figure 10:Total daily order lines per customer from 2022 - 2025-w13

2.3 Current forecasting procedure

To predict future demand, TKH Logistics directly copies the most recent estimates from the four-week order overviews provided by their customers if these are available. The customers and dates for which such overviews are not available are estimated by using the number of order lines from previous year on the same day of the week. Considering holidays with shifting dates is not ensured in the procedure, but team management does generally take it into account when making the staff schedules later, using their experience. For customer C the obtained value is multiplied by a growth factor of 1.10. For customers D and E, who are currently jointly estimated, a growth factor of 1.05 is used. The demand forecast is disaggregated to the customer level similar to the specifications provided for the deliverable for most customers. However, next to customers D and E, also customers N, O, and P are currently jointly estimated using an aggregated four-week order overview. As shown in Table 3, which summarizes the current method, order overviews are also provided and used for customers H, I, K, and L. For the final few days of the forecast horizon, there is generally not an estimate from the four-week overviews, as they are provided for whole weeks, such that on the Fridays in reality only a three-week forecast remains (the fourth week is the current week). In that case there is no estimate available. TKHL uses previous year demand then.

CUSTOMER(S)	METHOD OF ESTIMATION
A	Demand previous year
B	Demand previous year
C	Demand previous year * 1.10
D,E	Demand previous year * 1.05
F	Demand previous year
G	Demand previous year
H	Estimate four-week overview if available, else demand previous year
I	Estimate four-week overview if available, else demand previous year
J	Demand previous year
K	Estimate four-week overview if available, else demand previous year
L	Estimate four-week overview if available, else demand previous year
M	Fixed number estimate (200)
N,O,P	Estimate four-week overview if available, else demand previous year

Table 3: Current estimation method per customer

The current forecast is made in terms of order lines. Instead of predicting the number of order lines it might be more straightforward to predict the workload and required staff hours directly. Historic staff hours can however not be used to predict future hours directly as certain customers place orders that can be processed a few days before shipment. Staff hours data are polluted by the work done for different days. The rules for doing work on different days are not fixed and how much time is spent on work for another day is not recorded such that it is not possible to retrace how many staff hours were used for other days' work. Directly predicting required staff hours and the workload is therefore impossible with available data. Due to the lack of data to directly predict the daily workload, the best solution is to indirectly estimate the workload and required staff hours through demand. To indirectly predict demand, the number of shipments, order lines or order picks could be used. As shipments have very different order sizes, it does not provide great insight in the workload. Predicting the number of order picks or order lines provides relatively similar insight in the workload as both have fluctuating processing times per unit but can give a good insight into the total workload on a daily basis due to the larger daily volumes that limit the overall fluctuation of processing times on daily level. As the current forecast and customer order overviews are in order lines, we choose to stick to predicting the number of order lines, instead of picks.

2.4 Reconstruction of current forecast model

The current forecasting method is executed on Mondays, for one week ahead. To compare the deliverable to the current forecasting method the two models should deliver similar predictions in terms of aggregation level and horizon. For that reason, we reconstruct the current forecasting model conform the specified desired aggregation level and forecast horizon. To do that, the estimate for customers D and E has to be disaggregated. The same holds true for customers N, O, and P. In the first case this can easily be done as the previous year demand can be separated such that the same method of estimation can be used in the reconstructed model as in reality. However, customers N, O, and P cannot be disaggregated so easily as it is not possible to disaggregate the bundled four-week overview for these three customers directly. The reconstructed model therefore acts as if there is no four-week order overview for these customers as on the desired aggregation level there is in fact no such estimate. For customers N, O, and P we therefore also use historic demand directly, like for the other customers without order overviews. Besides this one difference between the forecasting level in reality and reconstruction, the current method could be followed directly for the reconstruction.

The reconstruction is made using the earlier composed dataset with daily number of order lines per customer and the four-week order overviews for the period 2023 to week 13 of 2025. The complete procedure for acquiring the reconstructed method is visualized in Figure 11.

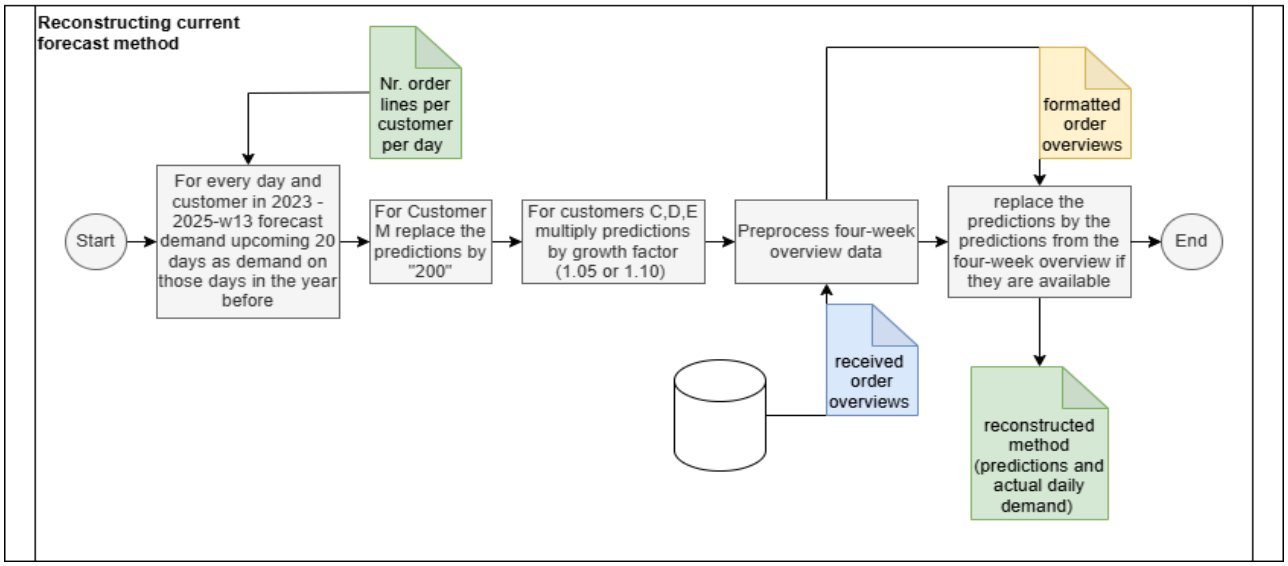


Figure 11: Steps to acquire reconstructed forecasting method

The resulting dataset provides for every workday within these 117 weeks the actual daily demand and forecasted demand on that date for the upcoming 20 working days. Lagging the predicted demand for X days in the future by X allows the direct comparison of actual demand with the prediction X days in advance, where X can be 1 to 20.

2.5 Performance of the current method

In S3.3 performance measures will be identified and proposed for model evaluation and comparison. These measures provide insight in the overall and customer level model performance. To create a benchmark to which we can compare method performance in Chapter 4, we calculate these measures for the reconstructed method over the test set that we will later introduce. The method should forecast a 20-day horizon. In the report we summarize the results using the mean of the metrics over a 7- and 20-day horizon. Seven days, as the staff schedules are made using the forecast from Wednesday for the week ahead, which is seven days. Twenty days, as this is the full horizon. The reconstruction results per forecast step are shown in *A.8 Reconstructed method test period results.*

If desired, refer to S3.3 for a more elaborate explanation and derivation of the performance measures. The results in Table 4 show that the Mean Weighted average Absolute Percentage Error increases as the prediction is made further in advance as the mean MWAPE over 20 days is larger than for 7 days. Analysing the MWAPE for each time step, we observe a large jump from the 17 to 18 day ahead prediction. the WAPE scores are all below 200%, except for December 31st and January 1st. December 31st the WAPE was around 400% for all prediction points. For January 1st, the WAPE was around 2,700% up to 17 days ahead and then jumped to almost 7,300%. The reason is that in 2025 January 1st was on a Wednesday, such that the provided order overviews from customers covered this day from 17 days in advance. So, for the 20-18 day ahead prediction, historic data was used instead. The large error around the new year shows that TKHL's current method seems unable to accurately predict demand during and around holidays. January 1st was responsible for a 41.5% increase of the MWAPE for days 1 to 17, and 112.3% for 18 to 20.

Looking at the bias in Table 4, the method seems to suffer from overestimation. This seems to be the result of the provided order overviews, as the predictions from different days ahead only differ because of the provided order overviews (i.e. the estimates are otherwise based on historic demand only which is the same regardless of when the prediction is made). The bias ranges from -13.6% for 15 days ahead to 3.6% for 20 days ahead. A plausible reason is that the provided order overviews are updated daily by adding and removing orders in the overview by the customer. Order estimates first increase due to new orders that are placed, ultimately leading to an overestimate. As the realization date gets closer certain orders are generally withdrawn such that the overestimation reduces again.

The mean absolute deviation shows a steady increase as the forecast is made further in advance, which makes sense as predicting further into the future is generally more difficult.

Overall	***Reconstruction***	
days ahead	**7 days**	**20 days**
MWAPE (%)	78.5	95.8
bias (%)	-8.5	-8.2
MAD	975.8	1208.8

Table 4: Overall performance reconstruction on a 7- and 20-day horizon

When we disaggregate the bias and MAD to customer level (A.8), we observe that over- and underestimation are a severe problem for several customers. The highest portion of forecast error comes from the estimates for customers C, I, K, L, M, and O, which are all customers with large order volumes. One particular observation is a sporadic large error for customers K and L for the predictions from 9 days ahead (*Figure 33 in A.8*). It seems that the provided overviews, which are used for these predictions, have a structural issue.

2.6 Seasonality assessment of demand

Effective demand forecasting requires objective understanding and the appropriate aggregation level. These were assessed in Chapter 1. It also requires the establishment of performance measures and the integration of the forecast in the end, which will be discussed in Chapters 3 and 5, respectively. The major factors next to level and trend that influence the demand should be identified to accurately predict future demand (Chopra, 2019). Certain models are more appropriate than others if demand shows seasonal patterns. The moving-average method does for example generally not provide sufficient results with seasonality (Winston, 2003). We explore the presence of seasonal patterns, and use the findings to identify, compare, and train time series methods in next chapters.

We analyse seasonality on three straightforward levels: within-week, weekly, and monthly. Within-week to assess potential seasonality within the week and the other two levels to assess the presence of seasonal patterns over the year. We do so by taking the mean demand from each "season" for every year and compare them by dividing each mean by the overall mean of the given year. In the main report the graphs are the aggregated demand.

The day of the week seems to affect demand. Tuesdays and Thursdays show lower demand than the other days as shown on the left in Figure 13. The exact pattern is not directly the same for all customers. Most customer demand does show a within-week pattern however (*A.9 Seasonality assessment of customer demand).*

A larger seasonal pattern, over the year, seems to be present as well. Looking at the weekly and monthly plots we observe that during the Dutch construction holiday demand is significantly lower in all years. Demand during Christmas and new-year seems to be lower as well. The period around the holidays seems to make up for the decrease in demand, as demand shows spikes there.

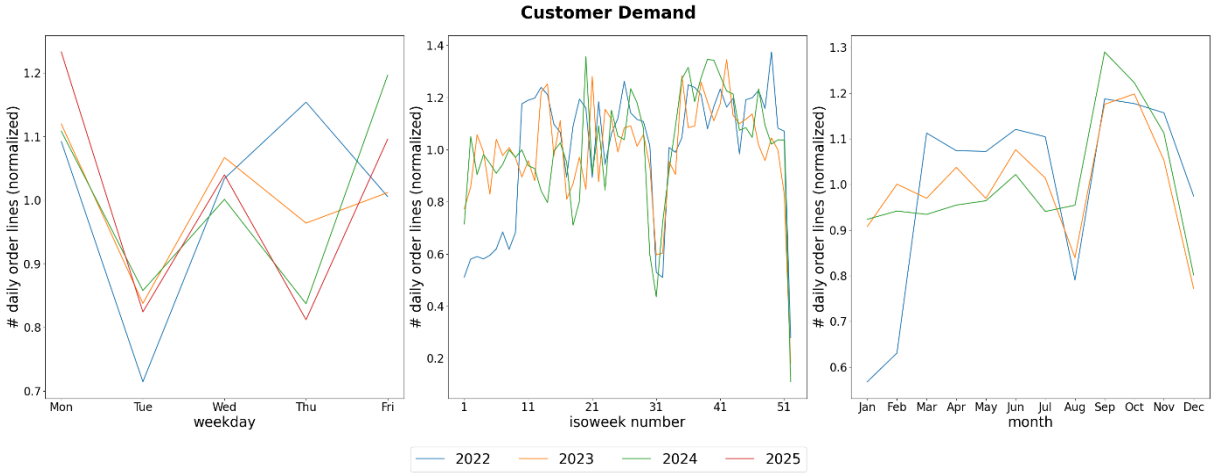


Figure 13: mean demand within-week, weekly, and monthly

Whilst analysing seasonality, the presumption that national holidays may affect demand gained ground. To better understand this potential effect on demand, we plot the mean demand during, the day before, and the day after holidays as well as the overall mean demand. Looking at Figure 14 we find that demand during national holidays seems to be significantly lower than usual. The day after the holiday seems to have lower demand on average as well, whereas the day before shows a slight increase. Results on customer level can be found in *A.9 Seasonality assessment of customer demand*. For all customers except customer P, it also seems that demand during a national holiday is lower than normal. For most customers, the holiday seems to affect the day before or after the holiday as well. The results tend to suggest that holidays may indeed affect daily customer demand locally.

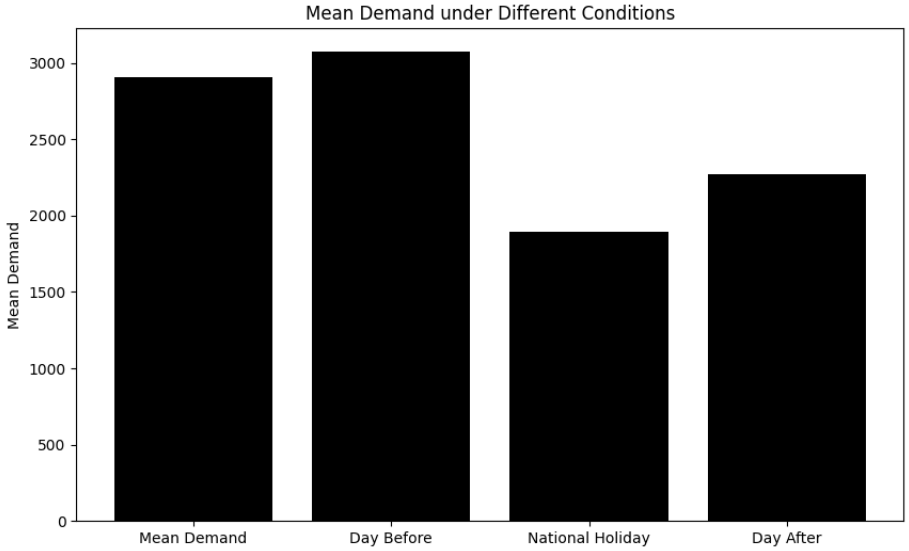


Figure 14: mean demand during and around national holidays

2.7 Seasonal-Trend decomposition using Loess

Removing level, trend and seasonality from a time series can be done using two different methods. These methods aim to create stationary data, which is data that do not show any trends or seasonality (anymore). In other words, fluctuations in the data are completely due to noise that is unexplainable with available data. One method called classical decomposition is to estimate the level, trend and seasonality components and subtract them from the data. This is a technique in which we try to remove noise by smoothing. Alternatively, we can compute differences between consecutive points using lag operators, which we refer to as differencing (Brockwell & Davis, 2016). To provide stronger foundation for the idea that there is seasonality both within the week and throughout the year we may try one of these methods and assess whether the residuals do contain white noise only.

Decomposing the data using within-week and yearly seasonality can be done using the classical decomposition method proposed by Holt & Winters (Chopra, 2019). This traditional and trivial method of using averages tends to work well on stable seasonal patterns. However, having seen the demand data of TKHL, the seasonal patterns are not that clear and robust, providing reason to explore more sophisticated and complex methods that can handle multiple seasonal patterns. One such method is Seasonal-Trend decomposition using Loess (STL) (CLEVELAND, 1990). In this method, a local weighted regression (Loess regression) replaces the more traditional moving average method allowing more complex patterns to be captured. We use this method to decompose demand assuming a within-week seasonality, and periodicity of a year such that both seasonal patterns, which we expect to be present after analysing the weekly and monthly plots before, can be captured.

Figure 15 shows how incorporating the two seasonal patterns using STL significantly reduces the noise in the demand. However, the residual does not yet represent white noise. Especially in 2023 the spikes are relatively large still. Part of the residuals can potentially be explained by the holidays. This holds for example for the spike in the residual at the beginning of 2024 that is likely caused by Christmas and the new year. We conclude that TKHL has to consider in particular holidays that have no fixed dates, as they do not do this consistently now, but it seems to affect demand. The results per customer show comparable results although slightly less robust with more outliers, which can be explained by the fact that disaggregated forecasts tend to be less accurate (Chopra, 2019). Decomposed customer demand can be found in *A.10 STL decomposition of customer demand.*

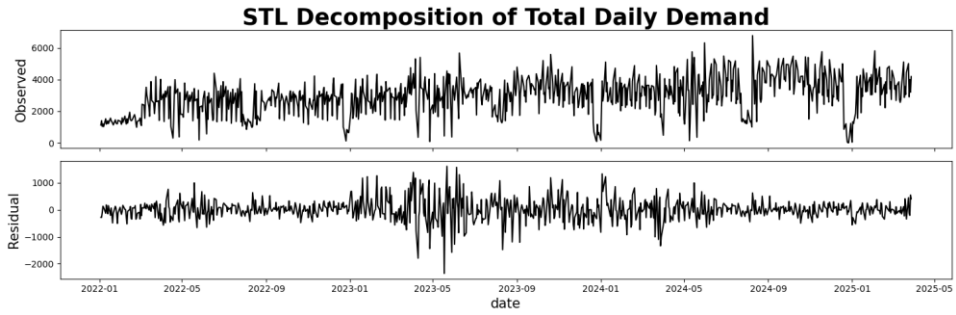


Figure 15: STL Decomposition of total daily demand

2.8 Conclusion

In this chapter, we analysed, reconstructed, and assessed the current forecasting method of TKHL. We analysed the presence of seasonal patterns and were able to reduce the noise in demand through incorporating the seasonal patterns in a STL decomposition of demand. We conclude that:

1. We have enough historic data on desired aggregation level to train a time series method
2. The current method is unable to accurately predict demand as it suffers from bias and large errors on both aggregate and customer level, especially during and around holidays and in particular when these holidays take place on different days over the years.
3. Yearly and within-week seasonal patterns, as well as holiday effects, are likely present in customer demand and have to be incorporated in the time series method.

When selecting the forecasting method in next chapter, we should ensure it can capture seasonal patterns and holiday effects. Explanatory variables may be required to handle these cases as well as to allow for the inclusion of order overviews and scrap orders, which we identified as an event that can be used to explain the two outliers in demand of customer F. Finally, customers K and L seems to provide order overviews that have a structural error for the prediction provided 9 days ahead, as the accuracy of these predictions is significantly lower than for the 8 and 10 day ago predictions. TKHL should share this finding with these customers to find out whether this issue is due to the way the estimates are composed, and whether it can be resolved to improve the quality of the prediction for 9 days into the future.

3. Literature review

In this chapter, we establish the method criteria and identify suitable methods in S3.1 based on the findings from Chapter 2 to answer RQ2.1. We identify model validation strategies in S3.2 and measures in S3.3 to answer RQ2.2. After this chapter, we are able to train a suitable time series method and benchmark its performance against the reconstructed method using proper validation techniques and metrics.

3.1 Time series methods with explanatory variables

In this section we will identify suitable time series methods by determining the method criteria in S3.1.1, different statistical methods in S3.1.2, and deep learning-based methods in S3.1.3. We conclude this section by presenting our findings in S3.1.4.

3.1.1 Method criteria and identification

Forecasting can be done using qualitative, time series, causal, and simulation methods (Chopra, 2019). Choosing which type is best can be hard. In this case however, we expect that future demand is related to historic demand, making time series methods more suitable than methods that are subjective (qualitative), assume highly correlated data with external factors (causal), or use simulation. Several types of time series methods will therefore be identified and analysed.

We concluded that seasonal patterns likely exist within the time series. We therefore want a method capable of capturing seasonal patterns. We want to assess whether the inclusion of the four-week order overviews improves predictability of demand. The method should therefore allow the inclusion of explanatory variables. A side benefit is that these methods also allow manually adding variables for holidays if we find that the inclusion of weekly and yearly seasonality can not capture this effect well, due to, for example, the fact that certain holidays have no fixed date. Finally, we want to do multi-step forecasting (forecast 20 days ahead), so the model should be capable of doing so.

Method criteria (hard): able to capture seasonal patterns, possible to include explanatory variables, and suitable for multi-step forecasting

---

The demand of most customers seems to be highly fluctuating in complex patterns and not be stationary, even after simple decomposition as done in S2.4. Models that can handle and capture non-linear and more complex patterns are therefore preferred. As we have to predict demand for 16 customers, a multivariate method may be beneficial as then only one model has to be trained, but this is no hard criterion. Demand between the customers is likely not or only slightly correlated, such that integration in a single multivariate model is likely not beneficial compared to combined univariate models when it comes to the prediction power (Korstanje, 2021).

Method criteria (soft): able to capture non-linear, complex patterns well and multivariate

---

In a systematic literature review on time series analysis with explanatory variables 30 methods were identified (Maçaira et al., 2018). The three most popular methods turned out to be regression models, artificial neural networks (ANN), and AutoRegressive Integrated Moving Average with eXogeneous inputs (ARIMAX). An autoregressive model uses a variation of linear regression to predict sequential data. One may thus argue that ARIMAX is rather a specific, it be very popular, type of regression model than a separate method type. Using that argument we may distinguish two popular methods; regression methods, of which ARIMAX is by far the most popular for time series analysis, and neural networks. Additionally, we may separate statistical methods, such as ARIMAX, from machine learning, and more specifically deep learning, based methods, like neural networks.

3.1.2 Statistical methods

(Seasonal) AutoRegressive Integrated Moving Average with eXogeneous inputs

As we established that the model should be able to handle seasonal patterns, we evaluate SARIMAX, instead of earlier identified ARIMAX, which also allows the inclusion of seasonal patterns within the time series and is one of the most popular statistical methods for time series forecasting currently. A SARIMAX model is build with several components including an AutoRegressive, Moving Average and Integration part composing the basic ARIMA model. This model is extended to capture seasonality using SARIMA. Considering outer factors can be done by the integration of exogeneous variables which we do by extending the SARIMA model to SARIMAX. To build the model, the data should be stationary, according to the Box-Jenkins method (Huang & Petukhina, 2022). This can be done through differencing, which we briefly touched upon in section 2.4, and also requires examination of autocorrelation plots and sometimes data transformations (Brockwell & Davis, 2016). The model order shall be determined to be able to fit the differenced series, and only then we may estimate the model after which the model can be assessed (Huang & Petukhina, 2022). Although SARIMAX is one of the most popular statistical methods for time series, the requirement of stationary data does require several steps before the model can be fit.

Prophet

Prophet is an open-sourced time series method released by Facebook in 2017 (Taylor & Letham, 2017). It shows impressive performance on data with additive trends and several seasonalities and has been widely used ever since (Kulkarni et al., 2023). Like SARIMAX, Prophet is a statistical method that approximates the relation between historical and future demand patterns (Pełka, 2023). The main benefit is that the Prophet model does a lot of work for the user, resulting in high user friendliness (Korstanje, 2021). To train the model, all that is required is a dataset with the dates and dependent variable ('y'). Multiple seasonal patterns and trends can be detected by the model as they are incorporated in the additive (or simple multiplicative) model. Seasonal patterns, like holidays, as well as other extra regressors, can also be added to the Prophet model manually, as Korstanje shows in his example to predict restaurant visitors. In this example he also suggests using two columns that contain windows around the holiday that might be impacted by it, like in our case the day before and after. Using grid-search the model hyperparameters can be tuned to optimize model performance. The trained model can do multi-step forecasting, but only for univariate data, so it requires a model for each individual customer.

3.1.3 Deep learning-based methods

Fully connected neural networks

Fully connected neural networks follow a schema where the input variables are provided in the input layer. Through X hidden layers, which basically multiply the inputs with weights and put them through activation functions, the derived output variables arrive at the output node. To train a connected neural network model, one requires a lagged dataset with the previous observations, as the sequentiality is not intuitive in a connected neural network. Next to that, the data should be standardized, using for example a standard or minmax scaler. Principal component analysis can be conducted to reduce the dimensionality of the data before training the actual model (Korstanje, 2021).

Recurrent neural networks (RNN)

As a solution to the required lagging of variables, recurrent neural networks were introduced. These have a feedback loop that makes them particularly suitable for time series analysis as the RNN can, through this loop, learn sequences. This makes multistep forecasting very intuitive as well.

Long Short-Term Memory Loss (LSTM)

LSTM is the most powerful recurrent neural network to do forecasting, in particular when seasonal patterns have longer intervals as this longer pattern can be captured with the long short-term memory part of the network (Korstanje, 2021). Again, to train the model, first the data has to be scaled and prepared, in this case by converting to a windowed dataset. Then the model training can take place. Although neural networks can sometimes obtain results much more powerful than classical methods, the downside is that training times are very long and tuning and parameterizing is difficult, for neural networks in general.

NeuralProphet

NeuralProphet improves the traditional Prophet by adding a neural network component for covariate and auto-regression modules, whilst it remains a user-friendly method (Kulkarni et al., 2023). In a case study towards the prediction of energy at a PV solar plant, the authors found that NeuralProphet, combined with ridge regression, was able to outperform other popular methods such as ARIMA, but also previously discussed LSTM, highlighting its effectiveness (Arias Velásquez, 2022). Compared to Basic Prophet, NeuralProphet can capture more complex patterns due to the feed-forward neural component and the allowance of adding lagged variables (Triebe et al., 2021).

The model consists of multiple modules that contribute additively to the overall forecast. These modules include trend effects (T), seasonality effects (S), regression effects for future-known exogenous variables (F), event and holiday effects (E), lagged observations of exogenous variables (L), and auto-regression based on past observations (A).

$$\hat{y}_t = T(t) + S(t) + F(t) + E(t) + A(t) + L(t)$$

The trend (T) is modelled as a piecewise linear function that is defined using a time-dependent offset ρ(t) and growth rate δ(t) which are estimated by fitting the model on the training data.

$$T(t) = \delta(t) \cdot t + \rho(t)$$

The seasonality module uses Fourier terms, analogous to its employment in Prophet (Taylor & Letham, 2017), to produce smooth seasonality effect functions. To model multiple seasonal patterns different Fourier terms can be defined per periodicity and the seasonal effect (S) is the sum of all the individual periodicities. Instead of additive contribution, the seasonality module can be configured to be multiplicative by multiplying the seasonal effect with the trend effect.

$$S^*(t) = \begin{cases} T(t) * S(t) & if\ S(t)\ is\ multiplicative \\ S(t) & if\ S(t)\ is\ additive \end{cases}$$

Future regressors represent variables for which both past and future values are known. Additive effects from future regressors (F) are modelled by fitting the coefficients (d) of the future regressors on the training data and taking the sum of the individual future regressor effects.

$$F(t) = \sum_{f \epsilon F} d_f * f(t)$$

Event effects (E) are captured analogous to future regressors, with each event as a binary variable that signals whether the event occurs or not. Like the seasonality effect, the future regressors effect and events effect can also be configured to be multiplicative.

The auto-regression module (A) in NeuralProphet predicts the target variables by regressing them against the past observations and is based on a modified version of AR-Net (Triebe et al., 2019), which enables multi-step forecasting. AR-Net is configured as a linear model but can be configured

with hidden layers in which case a fully connected Neural Network is trained. With both configurations, the model takes the p last observations (lags) as input and outputs predictions for h future time steps.

$$A^t(t), A^t(t+1), \dots, A^t(t+h-1) = AR\_Net(y_{t-1}, y_{t-2}, \dots, y_{t-p})$$

Lagged regressor effects (L), also known as covariates, are incorporated to model the relationship between other variables and the target time series. For each covariate, a separate lagged regressor module is created, to enable the model to individually attribute the effect of each covariate to the forecast. These modules operate similarly to the auto-regression module, with the difference being that the input consists of historical values of the covariate rather than the target variable itself.

$$L_x^t(t), L_x^t(t+1), \dots, L_x^t(t+h-1) = AR\_Net(x_{t-1}, x_{t-2}, \dots, x_{t-p})$$

Without the auto-regression and covariate components the NeuralProphet model is essentially the same as the classical Prophet model touched upon in Section 3.1.2 and only if these modules are used and include hidden layers the model incorporates neural networks for model training. If no hidden layers are configured classical linear regression is used.

3.1.4 Most appropriate models

To determine whether a statistical method, like SARIMAX, or a deep learning model shall be used for prediction we should know in which cases these models perform the best. Korstanje (2021) argues that from a theoretical perspective statistical methods may be preferred when the time series contains more information than the external variables and supervised models when the external variables alone can explain a lot. However, he concludes that the most reasonable thing to do is to use multiple models in a model benchmark (Korstanje, 2021). Instead of choosing one method, we therefore opt to try both a statistical and deep-learning method. From previous analysis, we learned that SARIMAX, as statistical method, and LSTM, as deep learning-based method are the most popular and state-of-the-art methods, but that these also both require extensive data preparation and tuning. We therefore introduced (Neural)Prophet as more user-friendly alternatives that show similar performance and are also conform the hard method criteria established before. Prophet is a more traditional additive statistical method, or rather a procedure, which is likely less capable of capturing complex patterns, but can be used as a benchmark to evaluate whether statistical or deep learning-based methods perform better on the given data. As NeuralProphet can also be configured like classical Prophet, it can be used to experiment with the impact of including neural networks for the regression components of the model.

3.2 Model validation strategies

As we are considering the implementation of advanced machine learning techniques, it is essential to consider how the different methods can be fairly validated and compared. A common risk with machine learning techniques is overfitting the model to the data. This implies that a model fits nicely to provided data but does not generalize well. When validating the model using data that is familiar to the model, as it was used when training, the results are biased, and results are likely better than when exposed to new data. To fairly compare the performance of the proposed method with the baseline reconstructed in previous chapter we require a proper strategy for model validation.

Korstanje (2021) proposes three strategies. The first strategy is to create a train-test split where approximately 20% up to 30% of the data is kept in a test set. The models may be fitted on the training data and are later evaluated and compared on the test data which is new to the model. In this method, the test performance matters, as they best replicate the future case in which new predictions are to be made that are unknown to the model.

Alternatively, another split can be added where you train data on a training set, benchmark model parameters on a validation set, and finally evaluate model performance based on a test set. This is appropriate when from different model variations one model has to be chosen that will finally be evaluated. Without the validation set (only train and test) it is difficult to first compare the model parameters and then evaluate the performance of the selected model, as comparison and evaluation would be based on the same data, creating bias. The downside of an additional validation set is that there will be less training data, which determines for a large part the model prediction power. To address the issue, Korstanje suggests retraining the selected model using both train and validation set after selecting the model variation based on the validation set.

When very little data is available it may not be affordable to keep a large fraction of data excluded from the training data. In this case cross-validation can be used to still get a good insight in model performance. K-Fold cross-validation is the most common cross-validation method (Korstanje, 2021) and fits the model k-times using a large training set that is evaluated on the remaining data as illustrated in Figure 16. For time series methods this type of validation has serious reliability risks as the estimates are generally based on trends and seasonality such that estimating a point in between two known periods is much easier than prediction of a point of which only past values are known. The time series split proposes a solution to this problem, in which only data available before the test data is used to train the model, as shown in Figure 17.

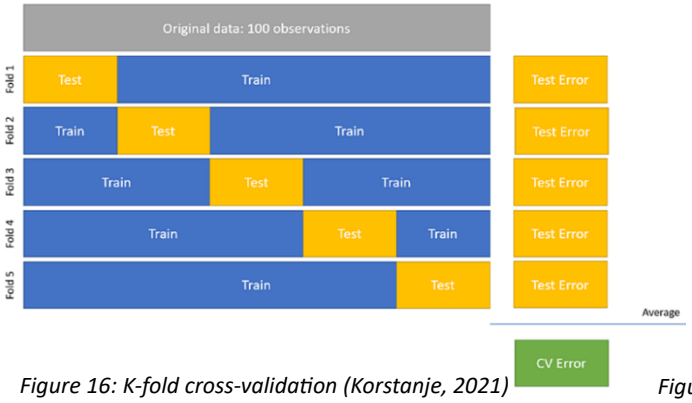


Figure 16: K-fold cross-validation (Korstanje, 2021)

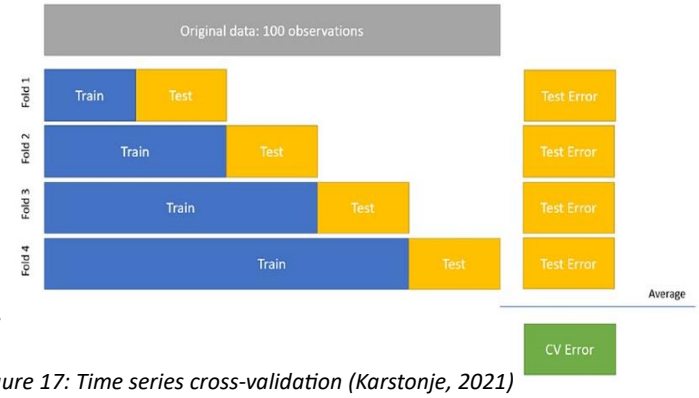


Figure 17: Time series cross-validation (Karstonje, 2021)

For most machine learning techniques, the test set can be a random selection of data. However, the particularity of forecasting is that the data is sequential. For that reason, it is more sensible to take the last portion of observations as test data as it is a closer replicate to future data (Korstanje, 2021).

3.3 Forecast accuracy measures

In this section we will identify popular performance measures in S3.3.1 and select and tailor these measures to our study needs in S3.3.2 to answer RQ2.2.

3.3.1 Measures in literature

The quality of the model outcomes should be evaluated systematically (Charles W, 1995). This will allow the comparison of the deliverable with the reconstructed model and provide insight in the overall quality of the models. Careful analysis of the forecast errors is required as managers use error analysis to determine the accuracy of the model (Chopra, 2019). We identify and establish popular measures that we can use for these purposes.

We define the forecast error (1) at period i in the same way we did in Chapter 1. The error (e_i) is the difference of the observed demand (X_i) and the forecasted demand (f_i):

$$e_i = X_i - f_i \qquad (1)$$

This error must be estimated at (or before) the moment the decision for which the forecast is used is made, as this is the accuracy of the forecast at the point decisions are made based upon it. The mean squared error (MSE) is one forecast error measure that relates to the error variance and is based on the principle that the random component of demand is distributed with a mean of zero and variance MSE (2). Therefore, it penalizes larger errors significantly more heavily than smaller errors. The MSE is mostly appropriate when the forecast error is symmetrically distributed around zero (Chopra, 2019).

$$MSE_n = \frac{1}{n}\sum_{i=1}^{n} e_i^2 \qquad (2)$$

If we take the absolute value of the errors and calculate its mean (3), we derive the mean absolute deviation (MAD). This measure is better than MSE in the case of a non-symmetric error distribution. Even in the case of symmetry MAD is appropriate when the cost of a forecast error is proportional to the error size (Chopra, 2019).

$$MAD_n = \frac{1}{n}\sum_{i=1}^{n} |e_i| \qquad (3)$$

Due to its lack of dimension, the mean average percentage error (MAPE), formulated as the average absolute error as a percentage of demand (4), is the most widely used measure for calculating the forecast error and preferred to the MAD (Charles W, 1995). It is a particularly good measure when underlying forecast deals with seasonal patterns and demand that varies over the periods as it considers the error in respect to its actual value such that the errors are put into perspective, in contrast to the MSE and MAD.

$$MAPE_n = \frac{1}{n} * \sum_{i=1}^{n} \frac{|e_i|}{X_i} * 100 \qquad (4)$$

A forecast may eventually no longer reflect underlying demand pattern due to uncaptured new events causing structural drops or increases in demand. In that case the errors will no longer fluctuate around zero. Accounting for this scenario is required and can be done using the bias (Chopra, 2019). The bias is formulated as the sum of errors as a percentage of total demand (5). When the error is random and unbiased, the bias should fluctuate around zero.

$$Bias_n = \frac{\sum_{i=1}^{n} e_i}{\sum_{i=1}^{n} X_i} * 100 \qquad (5)$$

3.3.2 Measure tailoring and selection

In Chapter 2 we evaluated seasonal patterns and concluded that these are likely present. Daily demand heavily fluctuates over the days, especially on customer level. The MAPE thus seems to be the best error measure. Bias shall also be considered to assess whether the forecasts reflect underlying demand patterns or suffer from systematic over- or underestimation of demand.

However, the MAPE poses issues as the MAPE can not be used when demand includes zero observations, as division by zero is not possible. Zero demand frequently occurs on customer level, so using the MAPE is not possible. The symmetric mean average percentage error (SMAPE) deals with this issue by dividing the absolute error by the average of the absolute values of the forecast and demand (6). This method works well when at least one of the two is not zero. If both are zero there is no error in reality, so we can handle the division error by putting zero instead.

$$SMAPE_n = \frac{1}{n} * \sum_{i=1}^{n} \frac{|e_i|}{\frac{|X_i| + |f_i|}{2}} * 100 \quad (6)$$

Although the SMAPE can handle the zero issue, it is still likely to overemphasize errors during periods with very low demand. Therefore, the MAD is recommended, instead of MSE due to likely non-symmetric error distributions and error cost proportionality.

The objective is to get a better total demand forecast, whilst distinguishing between customers. It is intuitive to measure the forecast error over the total demand, and not only separately for each customer. However, we should consider errors on customer level as we want customer separation. With this objective and requirement in mind, we can use a tailored version of the MAPE, which deals with the zero issue as well. We can (ignoring the zero-issue) calculate the absolute percentage error (APE) per customer (c) (4) and take the weighted average to get the overall APE. At period i we can formulate the weighted average absolute percentage error (WAPE[1]) of the customer demand by summing over the customers (C) (7). This deals with zero issues, as total daily demand is never zero.

$$WAPE_{C,i} = \sum_{c=1}^{C} \left(\frac{X_{c,i}}{\sum_{c=1}^{C} X_{c,i}} * APE_{c,i} \right) = \sum_{c=1}^{C} \left(\frac{X_{c,i}}{\sum_{c=1}^{C} X_{c,i}} * \left(\frac{|e_{c,i}|}{X_{c,i}} * 100 \right) \right) = \sum_{c=1}^{C} \left(\frac{|e_{c,i}|}{\sum_{c=1}^{C} X_{c,i}} \right) * 100 \quad (7)$$

We generalize the formula by taking the mean over all periods. We derive the formula of the Mean Weighted average Absolute Percentage Error (MWAPE) (8). We also take the weighted average for the bias (9). In addition, we select the MAD to understand the overall absolute error size of the model (10). For overall model performance evaluation, we use the following:

$$MWAPE_{C,n} = \frac{1}{n} * \sum_{i=1}^{n} \left(\sum_{c=1}^{C} \left(\frac{|e_{c,i}|}{\sum_{c=1}^{C} X_{c,i}} \right) * 100 \right) = \frac{1}{n} * \sum_{i=1}^{n} \left(\frac{\sum_{c=1}^{C} (|e_{c,i}|)}{\sum_{c=1}^{C} X_{c,i}} \right) * 100 \quad (8)$$

$$Bias_{C,n} = \sum_{c=1}^{C} \left(\frac{\sum_{i=1}^{n} (X_{c,i}) * Bias_{c,n}}{\sum_{c=1}^{C} \sum_{i=1}^{n} X_{c,i}} \right) = \sum_{c=1}^{C} \left(\frac{\sum_{i=1}^{n} X_{c,i}}{\sum_{c=1}^{C} \sum_{i=1}^{n} X_{c,i}} * \left(\frac{\sum_{i=1}^{n} e_{c,i}}{\sum_{i=1}^{n} X_{c,i}} * 100 \right) \right) = \frac{\sum_{c=1}^{C} \sum_{i=1}^{n} e_{c,i}}{\sum_{c=1}^{C} \sum_{i=1}^{n} X_{c,i}} * 100 \quad (9)$$

$$MAD_{C,n} = \frac{1}{n} * \sum_{i=1}^{n} \left(\sum_{c=1}^{C} |e_{c,i}| \right) \quad (10)$$

[1] The term “WAPE” is also mentioned in literature with other definitions, such as being the MAD divided by the total demand. Here we do not refer to this, or any other in literature existing, definition.

3.4 Conclusion

In this literature review, we introduced (Neural)Prophet along with other statistical and deep learning-based methods, as models that are suitable for our forecasting project. We learned that choosing between using a statistical or supervised model is difficult to do beforehand, which is why it is generally better to train multiple models and benchmark them against each other. We conclude that it is recommendable to train at least one statistical and supervised model and compare the results before determining which model to use in the end.

1. We suggest using Prophet and NeuralProphet as a statistical and deep-learning based method. These methods are conform the established criteria and the open-sourced methods developed by Facebook are user-friendly and able to provide relatively good insight in the procedure.
2. We evaluated different model validation strategies and learned that for time series analysis "time series cross-validation" is a particularly suitable strategy. We shall separate a test set from our data and use a validation set to determine model parameters. This way we can fairly validate and benchmark model performance using the test set.
3. We introduced different performance measures and concluded that the introduced MWAPE, weighted average bias, and MAD can be used to analyse and benchmark the models on overall level. On customer level we best evaluate performance using the bias and MAD.

4. Solution selection and validation

In this chapter we will explore how to predict demand and build the forecasting model. In S4.1 we decide upon the modelling approach and answer RQ3.1. In S4.2 to S4.5 we train and evaluate different model configurations. In S4.6 we compose and analyse the final model construction, answering RQ3.2 and RQ3.3. In S4.7 we present the conclusions of this chapter.

4.1 Model selection and approach

Model selection

In Chapter 3 we concluded that comparing the performance of a statistical and deep learning neural network with the reconstructed method is recommended, as it is data dependent which method works best. We proposed Prophet and NeuralProphet as two suitable and user-friendly methods. The NeuralProphet model provides the same basic model components as Prophet but introduces optional additional features such as local context through auto-regression, covariate modules, and neural networks (Triebe et al., 2021). NeuralProphet is build based upon Prophet and can be configured like Prophet by excluding the local context and neural network features. Therefore, we do not have to familiarize with two different frameworks, as Prophet is built on Stan (Carpenter et al., 2017) and NeuralProphet on PyTorch (Paszke et al., 2019). Instead, we only use the NeuralProphet framework but exclude the features unique to NeuralProphet for training the Prophet model, our selected statistical method. We first train this basic Prophet model and later expand to a NeuralProphet model by including autoregression and feed-forward neural networks. We use the NeuralProphet library in Python. The following model configurations, where the extensions build upon (and thus include the components of) the previous model, will be configured to analyse the impact of the different aspects:

S4.2 Base model: including trend and seasonality effects (Prophet)

$$\hat{y}_t = T(t) + S(t)$$

S4.3 1st extension: including holiday and event effects (Prophet)

$$\hat{y}_t = T(t) + S(t) + E(t)$$

S4.4 2nd extension: including four-week order overviews ((Neural)Prophet)

$$\hat{y}_t = T(t) + S(t) + E(t) \ (+ F(t) \ or + L(t))$$

S4.5 3rd extension: including (deep) auto-regression (AR) (NeuralProphet)

$$\hat{y}_t = T(t) + S(t) + E(t) + A(t)$$

Approach

In Chapter 3 we learned about data splits that ensure the model comparison results are not biased by the model training procedure. We use customer demand data up to 2024 for model training and hyperparameter selection by creating a separate validation set within this split. To evaluate model performance and perform benchmarks against the reconstructed method we use the collected data from 2025, which includes the first 13 isoweeks of the year, as a strictly separated test set. This will allow fair comparison of the methods as the demand during this period is new to all configurations. For realistic evaluation we use time series cross validation, which was introduced in Chapter 3.

We compare and benchmark the methods based on the performance measures selected in Section 3.3 for the predictions in the test set. The measures are:
- The bias and MAD to evaluate performance on customer level.
- The MWAPE, weighted average bias, and MAD to evaluate overall performance.

For each day and customer in the test set we make a forecast for the future 20 business. This implies, there will be 20 values per metric if we provide them for each step. As the staff schedules are made using the Wednesday forecast for the week ahead, which is a horizon of seven days, and we should not ignore part of the total forecasting horizon, we focus on the scores over a 7- and 20-day horizon.

4.2 Prophet with seasonality and trend effects

Configuration

In Chapter 2 we concluded the presence of a yearly and within-week seasonal effect. The basic additive Prophet model that includes the trend and the two seasonality effects has two components:

$$\hat{y}_t = T(t) + S(t)$$

Customer H is only a customer of TKHL since March 2024, so capturing a yearly pattern for this customer cannot be done as we would need at least two, preferably even more, records of the cycle (Hanke, 2005) so for this customer the seasonality effect only consists of a within-week periodicity.

As discussed in Chapter 3, the NeuralProphet framework offers options for multiplicative seasonal patterns as well. When the amplitude of the variation depends on the level, multiplicative seasonality is preferred (Gould et al., 2008). The amplitude of seasonal periodicity does not seem to change with the trend for all customers except customer I, L, O, and P (Figure 10 in S2.2). In 2022 demand was very low compared to the following years for customer I, and customer L, O, and P seem to have demand fluctuations that are slightly higher in years with higher demand. After deciding upon the model configuration and hyperparameters (*A.11 Basic Prophet model configuration and results*) we run the model with both additive and multiplicative seasonal effects and compare the results on the validation set to determine whether additive or multiplicative seasonality works best for these four customers. The results, as provided in *A.11 Basic Prophet model configuration and results,* show that the multiplicative models do not significantly change, or even decrease, performance. We therefore use an additive model configuration (as shown above) for all customers.

Results

The overall performance of the model configuration discussed in previous paragraph is estimated with the time series cross validation. The values for the MWAPE and MAD (Table 5) are higher than in the reconstructed model (Table 4 in S2.5) and we observe a relatively strong underestimation of demand.

Model components	***T & S***	
days ahead	**7 days**	**20 days**
MWAPE (%)	130.1	146.6
bias (%)	8.3	5.2
MAD	1523.4	1608.5

Table 5: Overall performance trend & seasonality model

Looking at the performance on customer level (*A.11 Basic Prophet model configuration and results*), we can explain part of the low overall performance. The prediction accuracy is the same or better for all the customers, except those that provide order overviews (H, I, K, L). The MAD of the last group is much lower in the reconstructed model. This indicates that to improve the model accuracy we must incorporate these order overviews. We observe strong underestimation of demand for nine customers when we look at the bias. Analysis of the predictions and results per time step (*A.11 Basic Prophet model configuration and results*) provides insight in the reason for this observation. The model does seem to capture most patterns, but has difficulties in finding the right magnitude, resulting in frequent underestimations on days with nonzero demand. Perhaps including lagged regressors to provide local context, which we will experiment with later in this chapter, will solve this

problem. Through this analysis we also observe that the model suffers from a large error in between Christmas and the new year. The model still seems to be unable to provide accurate predictions during holidays.

Although the introduced model with trends and seasonalities provides better predictions for the customers that do not provide order overviews, its performance runs short for customers providing order overviews compared to the reconstruction, suffers from very strong bias on customer level, and is unable to effectively handle holidays. In next subchapters we will extend this basic model to try solving the three issues we identified by including holidays and events, order overviews, and autoregression (to hopefully reduce the bias).

4.3 Prophet with holidays and events

Configuration

Holidays with shifting days are a significant problem in the reconstruction, but also in the previous method. Therefore, we assess whether the inclusion of Holidays as explanatory variables can solve this problem. The NeuralProphet framework allows the inclusion of exogeneous variables that are available both in the past and future without requiring autoregression and deep layers (so it remains basic Prophet). It refers to them as future regressors. Holidays can be added this way. Additionally, windows around the holidays, which are considered as separate events, can be incorporated. For example, we may add Christmas but as we noticed in Chapter 2 also demand around holidays seems to be impacted by the holiday. We want to try capture the direct effect around holidays using these windows. Too large values may lead to model overfitting, so we choose a window of one to limit this risk. We also add a window that fills the gap between Christmas and new year to capture sporadic demand due to customers that are closed during this period. We include Dutch national holidays (*Which Days Are Official Public Holidays in the Netherlands?*, n.d.) and the Dutch construction break per region. For three customers we include the Chinese new year and a window of one week on both sides, as they are closed during the two weeks around this period. Next to that, we learned in Chapter 2 that customer F had two outliers, pre-announced scrap orders, which we configure as event effects. The model composition looks as follows:

$$\hat{y}_t = T(t) + S(t) + E(t)$$

For customer H we do not include holidays for the same reason we do not include yearly seasonality: we have only one year of observations, so the model is likely going to overfit the effect. If more data becomes available over the years it can be added for customer H too.

Results

We obtain conspicuous predictions at first (*A.12 Results Prophet with holidays*). The model seems unstable, causing large prediction spikes. To improve model stability, we experiment with the batch size and learning rate. Theoretically, a small batch size makes the optimizer results noisy increasing the chance to bypass optima, whereas large batch methods tend to generalize poorly and suffer from overfitting (Keskar et al., 2016). The optimal batch size is likely a balanced value. Batch sizes are recommended to be a power of 2 for processing purposes. We compare the results of using a batch size of 64, 128, and 256 with the default batch size of 16 and compare learning rates 0.05, 0.1, 0.2, and 0.4 (we provide the results for one customer in *A.12*). We find that increasing the batch size to 128 smoothens the loss function, but with 256 the loss converges later, especially for the training set. A learning rate of 0.05 seems to be optimal for smoothening the learning curve but requires large computation times. We select a batch size of 128 and learning rate of 0.1 and conduct the time series cross-validation.

Adding the holiday and event effects significantly reduces the large spikes around the new year that we observed in the previous section and improves the overall model performance, showing significantly lower MWAPE and MAD than the model we started with (T&S effect), as shown in Table 6. The bias did not decrease but increase slightly for the 20-day horizon.

Model Components	*T & S*		*T, S & E*	
days ahead	**7 days**	**20 days**	**7 days**	**20 days**
MWAPE (%)	130.1	146.6	75.6	80.7
bias (%)	8.3	5.2	8.3	6.6
MAD	1523.4	1608.5	1281.1	1319.3

Table 6: Overall performance holidays extension

At customer level the mean absolute errors are further reduced compared to previous model. The model is however still outperformed by the reconstruction for the four customers that provide overviews and still suffers from large biases (*A.12 Results Prophet with holidays*).

Although the extension may not have completely solved the deficient performance around holidays, including holidays and a window of 1 around them significantly improved the test performance and seems to be a good extension of previous version. In further research, experiments with larger windows may result in even better performance, but we conclude that including the holidays and a window of 1 seems to solve most of the issue in this case. We continue extending the model to hopefully improve the performance of customers with order overviews and reduce the large bias that we still observed with this extension.

4.4 (Neural)Prophet with order overviews

Configuration

Previous model showed better prediction accuracy for all customers except the four customers that are predicted using order overviews in the reconstruction. Demand seems to follow the provided overviews better than can be estimated using historic demand only. However, the order overviews are also imperfect. We want to explore whether including the order overviews in the model can outperform the overviews on their own. We go from a strict extrapolation forecasting method, assuming that past patterns will continue in the future, to a more causal forecasting method, in which predictions are determined using independent variables. Like mentioned in Chapter 2, there also exists a bundled order overview for customers O and P next to the disaggregated overviews for customers H, I, K, and L. The overviews became available from October 25th, 2024. From the original training set starting in 2022 we select only the observations from this date on. This significantly reduces the training set size but is necessary as the model cannot handle that many missing values. Imputation techniques offer a solution to missing values but, due to the serious number of consecutive missing values (all observations until October 25th), still faces a serious challenge as simple forward or backward imputation will by no means provide good estimates for the missing values. We note that the limited amount of training data compared to before may affect the results but continue with the limited selection of training data. In addition, we turn off yearly seasonality and leave out holidays as the remaining training data is covering less than a year, implying that these effects would lead to overfitting.

Evaluating the available features of the NeuralProphet framework we propose two inclusion methods in the following paragraphs. We then compare the performance of these methods with the most trivial method: direct replacement of the model prediction by the most recently provided estimate when that is provided (as applied in the reconstruction). This method does not suffer from the lack of training data as it allows all data from 2022 to be used, in contrast to the other two methods.

Future regressors

A common method is to add the provided order overview estimate as future regressors, like in the energy load forecast example provided by NeuralProphet (Triebe, 2024). The overviews are then included as future regressor effects (F) in the model composition:

$$\hat{y}_t = T(t) + S(t) + E(t) + F(t)$$

The values of future regressors must be available in both the training and test datasets. This requires aligning the provided estimates with the corresponding time steps; an estimate for day t+i, provided on day t, must be lagged by i rows such that the estimate for day t+i provided on day t ($\hat{X}_{t+i|t}$) appears in the row corresponding to day t+i, like illustrated in Figure 18.

Day \ Lag	**1**	**2**	**3**	**...**
t	X̂(t|t-1)	X̂(t|t-2)	X̂(t|t-3)	X̂(t|...)
t+1	X̂(t+1|t)	X̂(t+1|t-1)	X̂(t+1|t-2)	X̂(t+1|...)
t+2	n.a.	X̂(t+2|t)	X̂(t+2|t-1)	X̂(t+2|...)
...	n.a.	n.a.	X̂(...|t)	X̂(...|...)

Figure 18: Visualization of data structure when including order overviews as future regressors

A limitation of this method is that it can not be used to forecast the 20-day horizon all at once. For example, if today is day t, the value in row t+2 and column 1 is not yet available, as it represents the estimate for day t+2 received on day t+1, which is tomorrow. To resolve this issue, we can train a separate model for each time step. For instance, demand on day t+1 can be predicted using the full set in Figure 18. For day t+2, the estimate in column 1 is missing, so we would exclude that column from the input features to predict the second time step. This procedure can be repeated for all time steps to be predicted. The one downside of this method is its computational cost. The method requires training 20 models per customer with an order overview. We have 6 customers for which we believe that they could benefit from including order overviews. Instead of 16 models we now need to train 130 models.

Lagged regressors

An alternative approach that does not require training a separate model for each time step is to use the provided overviews as lagged regressor effects (L), resulting in the following model composition:

$$\hat{y}_t = T(t) + S(t) + E(t) + L(t)$$

Lagged regressors are only available up to the point that the prediction is made and can be structured as illustrated in Figure 19.

Day \ Time step	**1**	**2**	**3**	**...**
t	X̂(t+1|t)	X̂(t+2|t)	X̂(t+3|t)	X̂(...|t)
t+1	n.a.	n.a.	n.a.	n.a.
...	n.a.	n.a.	n.a.	n.a.

Figure 19: Visualization of data structure when including order overviews as lagged regressors

Using this method combined with scalar lags (only including the most recent value of the regressor as input in the AR-net), we forecast demand over a 20-day horizon on day t by leveraging the most recently provided overview. Each column in Figure 19 corresponds to a specific time step. The AR-Net is expected to learn the appropriate associations, assigning a heigh weight to the regressor corresponding to its intended forecast time step, and minimal for the other time steps. This allows the model to utilize the overview information without requiring 20 separate models per customer if

the AR-Net can identify the correct associations. With this method we extent to NeuralProphet, as the AR-Net is unique to NeuralProphet and not available in classical Prophet.

Results

We proposed two compelling methods for including the order overviews. Using lagged regressors is likely only successful if the model can understand how to capture the information. A more established method, using future regressors, has the downside of increasing computation times due to requiring a model for each time step to predict but did provide superior results in other case studies. We compare the results of these two methods with simply replacing the prediction from the previous model by the provided estimate if that is available. The overall performance of the three methods in Table 7 shows that direct replacement by the overviews provides by far the best performance showing lower MWAPE, bias, and MAD over both horizons. It seems that the approach of using historic demand as the main input and trying to complement it with the overviews is worse than using the overviews directly, potentially because the model remains stuck in local optima and is unable to find the optimal weights for the overviews.

Method	*Direct Replacement*		*Overviews as F(t)*		*Overviews as L(t)*	
days ahead	**7 days**	**20 days**	**7 days**	**20 days**	**7 days**	**20 days**
MWAPE (%)	27.6	40.7	51.4	105.9	62.6	97.1
bias (%)	-0.2	-0.7	-0.6	-7.7	-5	-6.5
MAD	717.9	959.7	954	1598.1	1174.6	1529.6

Table 7: Overall test performance of the three different methods of including order overviews

If we analyse the results on customer level for the customers that provide overviews, we observe that using the order overviews directly yields the best performance for customers H, I, K, L. Customers O and P provide a bundled overview that cannot be disaggregated making direct replacement not possible. We observe that the results of the two other methods, in which the bundled overviews are included as regressors, perform worse than the results obtained with the previous model. The trend, seasonality, and holidays model outperforms the models with order overviews for customers O and P.

We conclude that direct replacement of the prediction from previous model when an estimate is provided by the customer is the best method out of the three, we explored for incorporating the provided order overviews in the model. If eventually enough order overview data becomes available, we should retry incorporating the order overviews in the model as lagged or future regressors as we can then also include yearly seasonal patterns and holidays creating more stable and accurate predictions over the whole horizon probably, which may change the results observed now. These effects could now not be incorporated in the two proposed method using the overviews as regressors due to the limited period for which order overview data is available. We conclude that for the deliverable, the best method for now is to use the method with seasonalities and holidays from section 4.4 by default and replace the predictions with estimates from the provided overviews if these are available. This significantly improves the performance of the deliverable.

4.5 NeuralProphet model with (Deep)AR

Configuration

Finally, we focus on dealing with the model bias. We evaluate whether utilization of the features in NeuralProphet can further improve the constructed model. In particular, we want to assess whether model accuracy can be improved by capturing local context through auto-regression and more complex patterns by including deep layers in the AR-Net. The model composition includes trend, seasonality, event, and auto-regression effects:

$$\hat{y}_t = T(t) + S(t) + E(t) + A(t)$$

When order overviews are available the model predictions are replaced by the provided estimates from the order overview as we learned in previous section that this yields the best performance compared to the other two methods we experimented with.

Autoregression

Customers that observe demand patterns that fluctuate locally could benefit from including autoregression. For customer E this may improve the predictions as with previous models the seasonality of this customer was captured well, but the predicted magnitude often stayed behind even though the magnitude seemed to gradually change over time. Autoregression provides the local context which may help solve this issue. The most important parameter is the AR order, which is the number of past values to be regressed over (Triebe et al., 2021). The default configuration for autoregression in NeuralProphet contains no hidden layers and works as a single layer neural network with the AR order as number of input nodes (P), and the number of forecast steps (H) as the number of output nodes, as illustrated in Figure 20.

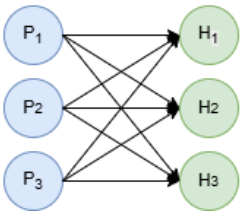


Figure 20: Single layer NN with 3 input and output nodes

Triebe (2024) suggests setting the AR order to at least the forecast horizon as this is preferable for the neural network. The autocorrelation function per customer *(A.14 ACF plot per customer)* shows that part of the lags of customers E, I, K, N, O, and P seem to be autocorrelated. Based on the ACF and suggestion from Triebe we choose to use 20 lags, the forecast horizon, and expect that the results for the six forementioned customers might benefit from the inclusion of autoregression.

Hidden AR-net layers

In the previous paragraph we discussed how the default autoregression is configured as a single layer neural network. This configuration can capture linear patterns between the included lags and the steps to be forecasted. More complex, potentially non-linear, patterns can not be captured this way. NeuralProphet provides an AR-net based AR module that can model non-linear dynamics through the configuration of hidden layers (Triebe et al., 2021). Triebe suggests that good enough performance can generally be attained without using these hidden layers, but to validate whether this claim holds for our dataset we explore the effect of using a simple feed forward neural network by including one hidden layer (I) with the same number of nodes as the input and output nodes as illustrated in Figure 21.

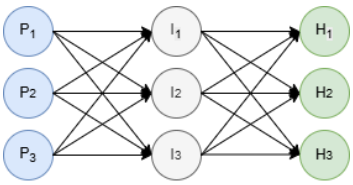


Figure 21: FFNN with one hidden layer and 3 nodes per layer

Results

We compare the results of including autoregression with and without a hidden layer with the best performing model from the previous section, which includes order overviews by direct replacement when these are available. The overall performance shown in Table 8 indicates that inclusion of autoregression for all customers harms the overall performance as all metrics got slightly worse for both AR and deep AR.

Overall	***Direct Replacement***		***A(t) - Linear***		***A(t) - Hidden layer***	
days ahead	7 days	20 days	7 days	20 days	7 days	20 days
MWAPE (%)	27.6	40.7	27.9	42.3	29.5	43.8
bias (%)	-0.2	-0.7	-1.7	-2.3	-1.3	-2.9
MAD	717.9	959.7	721.2	976.3	724.8	984.8

Table 8: Overall performance comparison between last model and inclusion of (deep) AR

If we look at the metrics on customer level (*A.15 Predictions NeuralProphet with (deep) AR*) we observe that for some customers the inclusion of (deep) AR did improve the results. From the two AR versions the model without a hidden layer worked best for customers A, D, F, K, and L based on the MAD and bias results of which at least one metric improved so much that we consider the trade-off (if present) to have a positive effect on the customer model performance. In addition, we also consider the AR version an improvement for customer C, even though the bias became 0.4% and 1.2% respectively, which is slightly worse but acceptable given the 7-day MAD reduction. Including a hidden layer, on top of adding autoregression showed impressive results for customer E in particular. The MAD was reduced by over 40% and the bias by over 80%. It seems that the expectation that providing local context could help improve the magnitude of the prediction for this customer was correct. The models of customers B and M also improved due to significantly lower bias, at the cost of a slight increase in the MAD. The inclusion of autoregression did not improve the results for the other customers. For the final model, we shall therefore use either no, linear, or deep layer AR based on these findings.

4.6 Final model analysis and results

Concluding this chapter, we discuss the results of the final model construction. The final model has the following components:

$$\hat{y}_t = T(t) + S(t) + E(t) + A(t)$$

A trend effect (T) is included for all customers. A within-week and yearly seasonality effect (S) are included, except for customer H that, due to a lack of data only has an included within-week periodicity. Dutch Holidays as well as customer-specific holidays and events are integrated as event effects (E). Linear auto-regression effects are included for customers A, C, D, F, K, and L and deep auto-regression effects for customers B, E, and M. If order overviews are provided for customers H, I, K, or L the predictions from the NeuralProphet model are replaced by the provided estimates.

The final model construction improved the MWAPE of the reconstruction by 64.8% and 57.0% on a 7- and 20-day horizon based on the result in the test set (Table 9). We were able to deal with most of the bias, as the final model showed bias of 1% or less for the horizons shown below. Finally, the overall MAD was reduced by 27.4% and 21.6%, respectively.

Overall	***Reconstruction***		***Final Model***		***Improvement final Model (%)***	
days ahead	7 days	20 days	7 days	20 days	7 days	20 days
MWAPE (%)	78.5	95.8	27.6	41.2	64.8%	57.0%
bias (%)	-8.5	-8.2	-0.6	-1	92.9%	87.8%
MAD	975.8	1208.8	708.5	947.7	27.4%	21.6%

Table 9: Overall performance comparison between final model and reconstruction

On customer level we observe that the final NeuralProphet model strongly outperforms the reconstructions for most customers that do not provide order overviews (*A.16 Results final model*). However, the model performance did not change much for the customers that provide order overviews, which makes sense as both the reconstruction and our proposal directly use the provided overviews when these are available. We observe that the MAD is still quite large for customers C, I, K, L, M, and O. This can be explained by the fact that these customers also contribute significantly to the total order volumes. In other words, these are large customers.

For most customers, the bias in the final model was strongly reduced. For customers A, B, G, and O we still observe very strong underestimation ranging from 37.4% up to 93%, however. We can explain the remaining bias for these customers by analysing the test period predictions (*A.16 Predictions final model*). We observe that these four customers provide low and mostly sporadic orders. The model struggles to predict their demand using the historical data, leading to mostly zero predictions and ultimately underestimation. Considering the order volumes for these customers we may argue that further research to variables that may reduce this bias is not time worthy considering the minor effect their demand has on staff and workload schedules, but for further model improvement it may be desired.

Improving the forecasting accuracy of the customers that still have relatively large MAD is likely more worthwhile. Four out of six large customers are providing order overviews, so the most straightforward method may be to try improving the accuracy of these overviews. We will further discuss and suggest this option in Chapter 5. In one of the interviews, it was mentioned that some fluctuation in the demand of customer C may be explained by the product arrival dates of sea containers, as there are generally backlogs on these products that should therefore positively correlate with demand. Explanatory variables like these, and potential others, may contribute to further enhancement of model performance.

4.6.1 Reflection on extend to which the norm is met

Although we changed the performance metrics and aggregate level throughout this research to better fit the objectives, the established norm for the forecast performance was set to 10% on aggregated daily volume in Chapter 1. To comment on whether we reached this norm, we evaluate the forecast performance of the final model on the test set in terms of the MAPE of aggregated demand. The average MAPE on a 7-day horizon in the test set was 13.8% (*A.16 Results final model*), slightly higher than established norm. However, notice how the MAPE of the reconstruction starts at 60.9% even though we estimated the MAPE in Chapter 1 to be approximately 30.8%. Part of the reason is that the new years period is included in the test set and that due to the low volumes during these days the recorded MAPE became very large, even though the absolute errors in terms of order lines were limited. The MAPE scores may for that reason be lower in periods without holidays as these events still slightly harm model performance, even with the final model construction.

4.7 Conclusion

The basic Prophet model constructed in this chapter suffered from bias, inferior performance around holidays, and unutilized information through the provided order overviews. In the following sections we tried improving the model by addressing these issues. The final model dealt with the issues mostly and showed significantly better performance than the reconstruction created in Chapter 2.

With the final model, bias still exists for some small customers. It could be explored whether methods to reduce these may improve method performance, but considering the small volumes of these customers (A, B, and G) the effect on the staff schedules is rather small such that careful evaluation of the cost-benefit should be conducted before spending time on this issue. Provided order overviews suffer from under- and overestimation as well but contain valuable demand information. In addition, we noticed odd behaviour in the provided order overviews of customers K and L for 9 days in advance. Somehow the error for the prediction 9 days ahead seems to be off frequently, whereas this issue does not occur for the 8 or 10 days ahead prediction of these customers. Perhaps a fundamental issue with the composition of the overview causes this error. We recommend TKHL to contact these customers to evaluate whether this is the case, and can be solved, to improve the quality of the overviews.

As directly replacing estimates by the provided order overviews seems to work the best now, TKHL should explore whether it can also obtain disaggregated order overviews for customers O and P instead of the bundled one it receives now. In addition, it may explore whether these overviews can also be provided by customers that do not do so yet. We will further discuss recommend steps regarding order overviews in S5.3 based on the finding that these heavily impact model performance.

As for the most important key takeaways, we conclude that:

1. The best model configuration for most customers includes trend, seasonality, event, and auto-regression effects but is ultimately customer-dependent. The best configuration found in this research is provided in Section 4.6 and the results were approximately 15% to 70% better in terms of the MAD compared to the reconstructed method for customers that do not provide order overviews. For customers that provide order overviews, the results are very similar, as they both directly use the provided estimates.
2. Incorporating holidays and a window of 1 around them improves model performance during these events. Further experiments with the window size may yield even better performance.
3. Order overviews contain valuable information for predicting customer demand but are not completely accurate and available. Therefore, it is recommended to explore improving and extending the width and accuracy of provided order overviews to further improve the model.
4. Including the information from order overviews can currently best be done by direct replacement of the prediction in case an estimate from the order overview is available. If eventually multiple years of data are available for the order overviews, this conclusion should be re-evaluated as incorporation through future or lagged regressors may be better then.
5. Including (deep) AR improved the performance for certain customers and dealt with part of the large bias. However, not for all customers. This shows that it is case dependent whether autoregression and hidden layers yield better performance. We note that further exploration with the hidden layer nodes and depth, as well as the number of lags, might further improve performance, but that the analysis done on the validation set resulted in acceptable results.
6. To further improve model performance the focus should be on improving performance for the larger customers (considering the cost benefit) and may be done through adding additional explanatory variables such as the arrival dates of sea containers for customer C.
7. The norm for the aggregated MAPE of at most 10% on a 7-day horizon was not exactly met, but close enough to consider the results of the final model acceptable.

5 Implementation plan

To answer RQ4, we distinguish between three implementation aspects. In S5.1 we discuss the deployment of the model in the online environment of TKHL; the technical implementation. In S5.2 we consider employee onboarding and training by reviewing change management literature. In S5.3 we argue that integration across the supply chain is required to further improve the model performance and provide the required next steps to realize this. In S5.4 we conclude Chapter 5 by providing our main findings regarding the model implementation.

5.1 Technical implementation

The reason for building the demand forecast in this research is that the staff and workload scheduling at TKHL suffers from a lack of trusted and accurate information of future demand. The introduced forecasting method based on (Neural)Prophet provides more accurate demand forecasts for the customers of TKHL. To benefit from this improved method, we consider the technical implementation of the model to ensure the method can be used.

The proposed forecast method should be trained and run iteratively on a daily basis. To do so without much manual intervention we consider the technical implementation of the method. To understand the required input, output, and processes of the method we visualize the configuration in Figure 22.

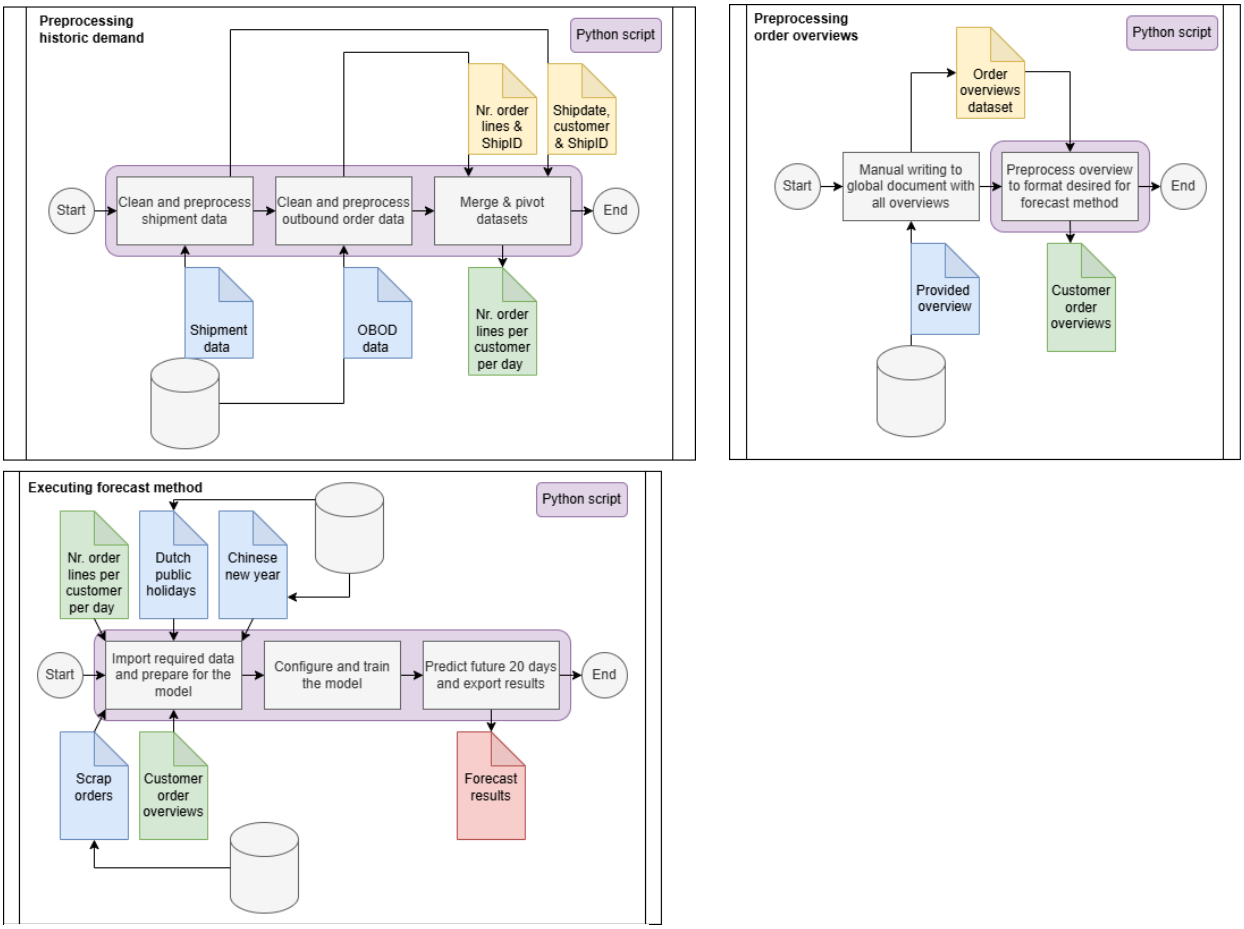


Figure 22: Technical configuration proposed forecast method

To implement the model in practice three python scripts have to be run using in total six different input files. The two holidays files only have to be updated when the holidays of the current year are no longer covered. The scrap orders file should be updated when a new scrap order is scheduled. The remaining three files are to be updated at the end of every day so they contain the information for all days including the current day, as the model uses all available information for training the model that is used to predict the future 20 days. The python scripts can be run automatically at a given time, which should be at the end of a day when the totals of the current day are known. To run the model automatically, it should be able to access the files illustrated in blue in Figure 22. The holidays and scrap orders require manual periodic updating whereas all other input files can be extracted from the datawarehouse and do not require manual intervention. Using Figure 22 TKHL can determine how the procedure, that takes approximately 20 – 30 minutes, can be executed on a daily basis.

5.2 Change management

> "A vision can only fully unfold its power if all the people concerned not only know it but also understand what significance the vision has for their everyday work" (Stolzenberg & Heberle, 2022).

The vision underlying this research is that a good operations environment can only be achieved when there is a level of control and calmness around the scheduled capacity and workload. We suggest holding three session types to establish this vision and control the implementation of proposed method. We provide practical concepts for the sessions in *A.17 Change management meeting concepts*. The change initiators are responsible for hosting the sessions and the team management should attend the sessions as well to show they support the vision.

A vision and method information session in which the vision and the forecasting method functionalities and instructions are discussed is required to enable staff to work with the method and understand the underlying vision. In the session the vision background, vision implication, forecasting method, and future steps should be discussed. At the end of the session a panel discussion is held.

To establish the vision the people concerned with the forecast should understand the significance of the vision and method. This can be done through a vision dialog kick-off in which the practical value of the vision can be made clear (Stolzenberg & Heberle, 2022). During the kick-off, the team discusses the vision, its relation to their work, and can be used to present case scenarios in which through interactive participation the value of the proposed model can be illustrated. A concept of using such case scenario in a simulation game to illustrate the value of the model is provided in *A.17 Change management meeting concepts* as well.

The exchange on the vision implementation, progress, and challenges should be maintained using continuous vision dialogs to keep the vision alive (Stolzenberg & Heberle, 2022). These sessions may be held every few weeks at first and should provide insight in the progress of the implementation and whether changes are required. During the sessions model users (staff) can be trained to understand the value, strengths, and weaknesses of the model. The team should ultimately recognize that the predictions serve as an important source of insight to inform their decision making. However, the predictions should be approached with discernment, as they provide projections rather than definitive truths. Stolzenberg & Heberle (2022) provide several process ideas for this continuous vision dialog and suggest it requires approximately an hour.

5.3 Supply chain integration

In Chapter 4 we observed the prediction power of the provided order overviews. This finding highlights the importance of close collaboration with the TKHL partners to obtain information that can serve as powerful input in the forecasting method. Chopra (2019) suggests that this collaboration takes time and effort but that the benefits generally outweigh its costs. To realise the integration across partners, a cross-functional team may be recommended (Chopra, 2019). The most important finding is that order overviews such as the ones that few customers already provide significantly improve forecast accuracy compared to methods that use historic data as its main input. With that, TKHL should invest time in exploring whether:

1. The quality of the already provided order overviews can be further improved.
2. Customers that do not provide these order overviews (on desired customer level) can start providing these overviews as well.

Obtaining these estimates and if possible, improving the quality of them will help further improve the demand forecast provided in this research and including order overviews for customers that do not yet have them within the provided method can be done easily by extension of the current model. We suggest to:

1. Establish performance measures, like the ones introduced in this work, to monitor performance of the currently provided overviews and steer towards improving their accuracy.
2. Explore the possibilities of obtaining order overviews on customer level for the customers that do not yet do this.
3. Explore the possibilities of obtaining other information that may help accurately predict future demand.

In addition, we found (what seems to be) a structural error for the 9 day ahead forecast of customers K and L and recommend sharing this finding with these customers to find out whether this issue can be resolved. When this issue can be resolved it will also improve the model performance for these customers.

5.4 Conclusion

To solve the core problem identified in Chapter 1 the implementation of the forecast model is an indispensable aspect as the forecast model alone will not solve the problem as the fact that the current model is not being used either shows. We discussed three aspects of the implementation. The technical implementation is necessary to be able to configure predictions everyday without much manual interventions and Figure 22 in S5.1 will allow TKHL to determine the best steps for the actual technical implementation. To create trust in the model and train staff to work with the method we dove into change management and proposed three session types that will ensure and guide the vision change. Finally, to further improve the model performance we concluded that supply chain integration is necessary, so we proposed steps to take for TKHL to improve their integration across their supply chain. We conclude that for a successful implementation of the method TKHL should:

1. Integrate the proposed final forecasting method in the online environment of TKHL.
2. Participate in change management to allow staff to trust and be able to work with the method in a way that will ultimately realise the underlying vision.
3. Invest in supply chain integration to further improve method performance.

6 Conclusions and recommendations

In this closing chapter, we derive the overall conclusions of our research which aims to reduce the mismatch between staff and workload schedules at TKH Logistics in Haaksbergen by providing a better demand forecast and implementation plan in S6.1. In S6.2 we provide recommendations for TKHL regarding the next steps for model improvement, implementation, and further research. S6.3 consists of a research discussion and limitation evaluation.

6.1 Conclusion

The mismatch between daily workload and capacity at the order picking division of TKHL is caused by the staff and workload schedules that are inadequately aligned. The main underlying reason is the lack of trust in, and reliability of, the current demand forecasting method employed at TKHL. The research set out to provide a reliable demand forecast and an implementation plan that ensures the proposed method is trusted and used, ultimately leading to the realisation of underlying research vision that a good operations environment can only be achieved when there is a level of control and calmness around the scheduled capacity and workload.

1. The current forecasting method has severe flaws caused by the lack of a professional method that can handle seasonality, including holiday effects, and utilize historic demand.

We identified suitable time series methods, validation techniques, and performance measures. A basic Prophet model configuration with trend and seasonality effects showed improvements for most customers but still had three main flaws. It suffered from large bias, ignored useful information from provided order overviews, and still showed deficient performance around holidays. Dealing with these issues improved the model performance and gave the following insights:

2. Including local context through (deep) autoregression enhances model performance, but its impact is customer-dependent. The MAD and bias could be reduced by over 40% and 80% respectively for one customer, whereas its impact was minimal for certain other customers.
3. The professional additive model that captures trend through a piecewise linear function, seasonality using Fourier Terms, events as binary variables, direct usage of provided estimates, and local context through auto-regression using (deep) AR-Net strongly outperforms the current method. On a 7-day horizon the MWAPE was reduced by 64.8%, the bias by 92.9%, and the MAD by 27.4%. On a 20-day horizon the results were 57.0%, 87.8%, and 21.6% respectively. Compared to the basic trend and seasonality effect model, the outperformance is even greater, suggesting that the final NeuralProphet model is preferred to more basic and classical methods that only capture trend and seasonality effects. The outperformance can be attributed mostly to the integration of special events and provided order overviews. Local context through AR had a minor effect on the overall performance.
4. The norm of no bias and a MAPE of 10% on daily aggregated level over a 7-day horizon was not obtained by a MAPE shortcoming of 3.8 percent points but deemed good enough, given the test results of the reconstructed method. The configured method should replace the current method and provide the input for staff and workload scheduling decisions.
5. Model predictions more than 17 days ahead should be trusted less, mostly due to the lack of estimates from customer order overviews for more than 17 days ahead.
6. The technical implementation is required to be able to run the method on a daily base. Vision dialogs are required to familiarize staff with the underlying vision and their role and responsibility in its realisation. Supply chain integration is necessary to further improve model performance.

6.2 Recommendations

Regarding the proposed model and its implementation, we recommend TKH Logistics to:

1. Consider the three aspects of implementation discussed in Chapter 5. This allows obtaining the predictions without manual intervention, shows the value and opportunities for further improvement of the method and vision to staff, and manages supply chain integration, which is desired for further model development. Regarding the supply chain integration, TKHL should also:
    a. Gain insight in the (what seems to be a) structural error for the 9 day ahead estimates in the order overviews from customers K and L to hopefully resolve that issue.
    b. Explore the possibilities of obtaining the order overviews on disaggregated level for customers O and P, as we learned that it is now not possible to effectively use the provided bundled overview.
2. Explore with the integration of more explanatory variables, such as arrival dates of sea containers for customer C, and evaluation of different holiday windows to further improve the model. (Note that we expect that most improvement can be obtained by focusing on the model performance for the larger customers, of which most provide order overviews. The main focus should therefore be on the supply chain integration and improvement of order overview quality.)
3. Evaluate whether aggregating the forecast for the small customers is possible as it may improve the prediction accuracy because predictions for some small customers still suffer from large bias (underestimation) in the final model due to the low and sporadic demand.
4. Distinguish model performance between different horizons. Especially from day 17 onwards the model performance reduces significantly, due to the frequent lack of provided estimates. Model performance depends on the time-step. This behaviour should not be forgotten when interpreting the model results.
5. Monitor and control the model performance by establishing a model KPI. The MWAPE is suitable for that purpose.
6. Re-evaluate whether directly copying order overviews is the best method for its inclusion in a few years, as integrating them as explanatory variables can then also be done whilst capturing yearly seasonal patterns, which was not possible now due to the lack of order overview data.

Regarding data collection, we learned that the available data does not allow directly forecasting the required staff, mostly due to inaccurate pick time measurements. Eventually, directly predicting the required number of staff may be preferred. To allow this type of prediction TKHL should:

7. Re-evaluate the way of collecting pick times, to ensure the measurements reflect reality well.

We also provide two recommendations that do not directly refer to the proposed method and its implementation for the order picking division but consider a wider scope. To fully solve the problem, we think it is necessary/preferable to:

8. Establish the internal processing rules (*A.6 Internal processing rules*) as these are now frequently bypassed even though they can ensure orders are processed in time. This could be done by directly converting the demand forecast to a workload schedule based on the rules. The workload schedule can then be used to derive a staff schedule. This can be done by using order line pick time estimates, like provided in S1.3.6, to convert the workload schedule to the required staff hours. To do this, TKHL (like mentioned before) needs to look into improving the measurement method of pick times as these measurements do not reflect reality now. In addition, we recommend to conduct further research towards optimal workload and staff scheduling

procedures, using the demand forecast predictions, as the current internal processing rules are likely suboptimal for creating aligned workload and staff schedules.
9. Follow the procedure in this research to build a similar forecasting model for the incoming product flow to serve as input for the inbound order division.

6.3 Discussion and research limitations

The research required certain decisions and assumptions to be made that led to discussion points and research limitations.

1. The ideal method for solving the action problem would have been to directly predict the workload and demand, however historic data for these purposes are not available. This was an important limitation for deciding upon the research approach and had this data been available the deliverable would likely have been a forecast that directly estimates the required staff. This was not possible and as the next best thing to predict was the demand in terms of order lines (or picks), we did that in this research.
2. The constructed historic demand dataset should reflect reality well. However, we learned that delivery dates are sometimes manually overwritten due to maximum daily lettings and inaccurate original shipping dates. We believe this affected only a small data portion and that the way of handling these issues deals with (most of) the wrong date entries but this cannot be guaranteed. This poses a potential issue regarding data reliability.
3. The research focused on time series methods. For the customers that provide order overviews this may not be the preferred method as we learned that in reality these order overviews seem to provide more information than historic demand.
4. The configuration of a basic Prophet to eventually a NeuralProphet model that utilises deep autoregression involved hyperparameter tuning, including holiday windows, and experimentation with different methods of including order overviews. Further tuning and experimentation with other methods may further improve the results and affect the research findings, even though we believe that sufficient experimentation was conducted to obtain reliable and sufficient results.
5. Time series cross-validation should provide unbiased results given the separated test set and ensure the validity of the results. We do note that the test period of 13 weeks is quite small (one quartile) which may make the results relatively unstable. This decision was made to ensure the model had training data from three complete cycles which should provide better results. The short validation period also ensures the results likely represent results for actual deployment well, as recent demand is likely more representative for current demand.

References

Arias Velásquez, R. M. (2022). A case study of NeuralProphet and nonlinear evaluation for high accuracy prediction in short-term forecasting in PV solar plant. *Heliyon*, *8*(9). https://doi.org/10.1016/j.heliyon.2022.e10639

Azevedo, A. , & S. M. F. (2008). *KDD, SEMMA and CRISP-DM: a parallel overview*.

Bhanja, S., & Das, A. (n.d.). *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting*.

Brockwell, P. J., & Davis, R. A. (2016). *Introduction to Time Series and Forecasting*. http://www.springer.com/series/417

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Charles W, C. Jr. (1995). Measuring forecast accuracy. *The Journal of Business Forecasting Methods & Systems*, *14*(3), 2–2.

Chopra, Sunil. (2019). *Supply chain management : strategy, planning and operation*. Pearson Education.

CLEVELAND, R. B. (1990). STL : A seasonal-trend decomposition procedure based on loess. *J Off Stat*, *6*, 3–73. https://cir.nii.ac.jp/crid/1571417124982951296

Girshick, R. (2015). *Fast R-CNN*. http://arxiv.org/abs/1504.08083

Gould, P. G., Koehler, A. B., Ord, J. K., Snyder, R. D., Hyndman, R. J., & Vahid-Araghi, F. (2008). Forecasting time series with multiple seasonal patterns. *European Journal of Operational Research*, *191*(1), 207–222. https://doi.org/10.1016/j.ejor.2007.08.024

Hanke, J. E. and W. D. W. (2005). *Business forecasting*.

Harvey, A., & Ito, R. (2020). Modeling time series when some observations are zero. *Journal of Econometrics*, *214*(1), 33–45. https://doi.org/10.1016/j.jeconom.2019.05.003

Heerkens, Hans., Winden, A. van., & Tjooitink, J.-Willem. (2021). *Solving managerial problems systematically*. Noordhoff Uitgevers-Groningen.

Huang, C., & Petukhina, A. (2022). *Statistics and Computing Applied Time Series Analysis and Forecasting withhPython*.

Jafari, R. (2022). *Hands-On Data Preprocessing in Python : Learn How to Effectively Prepare Data for Successful Data Analytics*. Packt Publishing, Limited. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=3125175

Jamal, P., Ali, M., Faraj, R. H., Ali, P. J. M., & Faraj, R. H. (2014). 1-6 Data Normalization and Standardization: A Technical Report. In *Machine Learning Technical Reports* (Vol. 1, Issue 1). https://docs.google.com/document/d/1x0A1nUz1WWtMCZb5oVzF0SVMY7a_58KQulqQVT8LaVA/edit#

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., & Tang, P. T. P. (2016). *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*. http://arxiv.org/abs/1609.04836

Korstanje, J. (2021). Advanced forecasting with python: With state-of-the-art-models including LSTMs, Facebook's prophet, and Amazon's DeepAR. In *Advanced Forecasting with Python: With State-of-the-Art-Models Including LSTMs, Facebook's Prophet, and Amazon's DeepAR*. Apress Media LLC. https://doi.org/10.1007/978-1-4842-7150-6

Kulkarni, A. R., Shivananda, A., Kulkarni, A., & Krishnan, V. A. (2023). Time Series Algorithms Recipes. In *Time Series Algorithms Recipes*. Apress. https://doi.org/10.1007/978-1-4842-8978-5

Loshchilov, I., & Hutter, F. (2017). *Decoupled Weight Decay Regularization*. http://arxiv.org/abs/1711.05101

Maçaira, P. M., Tavares Thomé, A. M., Cyrino Oliveira, F. L., & Carvalho Ferrer, A. L. (2018). Time series analysis with explanatory variables: A systematic literature review. *Environmental Modelling & Software*, *107*, 199–209. https://doi.org/https://doi.org/10.1016/j.envsoft.2018.06.004

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. http://arxiv.org/abs/1912.01703

Pełka, P. (2023). Analysis and Forecasting of Monthly Electricity Demand Time Series Using Pattern-Based Statistical Methods. *Energies*, *16*(2). https://doi.org/10.3390/en16020827

Ramu, G. (2022). *The ASQ certified six sigma yellow belt handbook* (Second edition). ASQ Quality Press. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=3508514

Smith, L. N. (2015). *Cyclical Learning Rates for Training Neural Networks*. http://arxiv.org/abs/1506.01186

Stolzenberg, K., & Heberle, K. (2022). Change Management: Successfully Shaping Change Processes - Mobilizing Employees Vision, Communication, Participation, Qualification. In *Change Management: Successfully Shaping Change Processes - Mobilizing Employees Vision, Communication, Participation, Qualification*. Springer. https://doi.org/10.1007/978-3-662-65396-8

Subasi, A. (2020). Introduction. *Practical Machine Learning for Data Analysis Using Python*, 1–26. https://doi.org/10.1016/B978-0-12-821379-7.00001-1

Taylor, S. J., & Letham, B. (2017). *Forecasting at scale*. https://doi.org/10.7287/peerj.preprints.3190v2

Triebe, O. (2024). *NeuralProphet*. Https://Neuralprophet.Com/.

Triebe, O., Hewamalage, H., Pilyugina, P., Laptev, N., Bergmeir, C., & Rajagopal, R. (2021). *NeuralProphet: Explainable Forecasting at Scale*. http://arxiv.org/abs/2111.15397

Triebe, O., Laptev, N., & Rajagopal, R. (2019). *AR-Net: A simple Auto-Regressive Neural Network for time-series*. http://arxiv.org/abs/1911.12436

Which days are official public holidays in the Netherlands? (n.d.). Https://Www.Government.Nl/Topics/Working-Hours/Question-and-Answer/Public-Holidays-in-the-Netherlands.

Winston, W. L. (2003). *Operations Research - applications and algorithms*. www.duxbury.com

Appendices

A.1 Warehouse process description

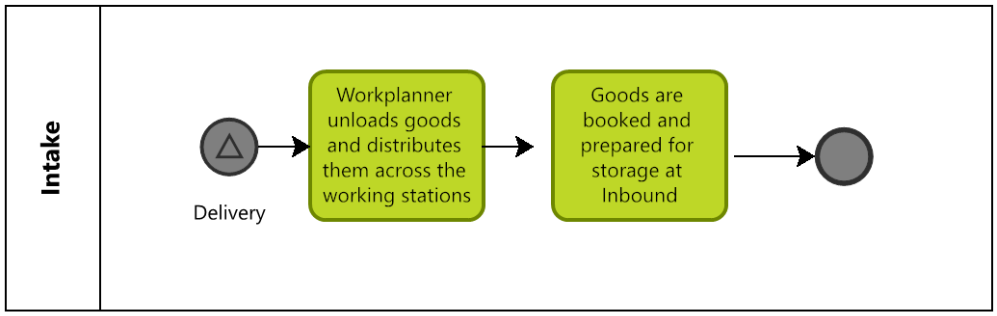


Figure 23: Inbound process

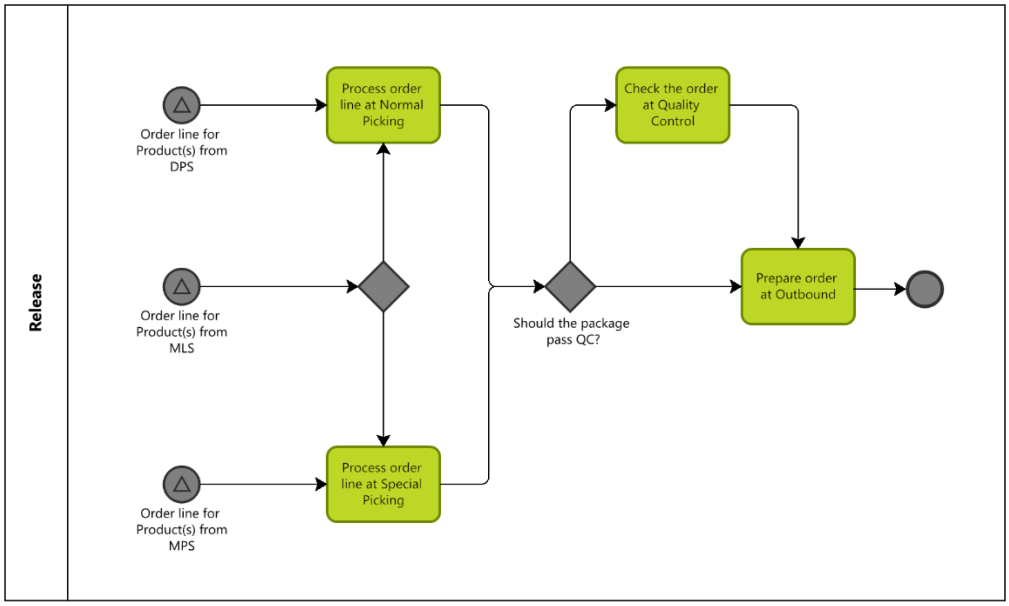


Figure 24: Outbound process

A.2 Pearson correlation test

The data from 2024 with the total number of order lines per day and the average number of processed order lines per staff hour on that day were assessed for correlation using the Pearson correlation test. The results are as follows:

	NP	NP
Alpha	0.01	0.01
Pearson correlation (r)	0.756454389	0.618936764
Degrees of freedom (df)	253	253
t-value	18.3964666	12.5340976
t*	2.34	2.34
Conclusion	Reject H0: r=0	Reject H0: r=0

Table 10: Results Pearson correlation test for productivity and daily order lines

In addition, visualisation of the two variables in a scatter plot shows the correlation visually:

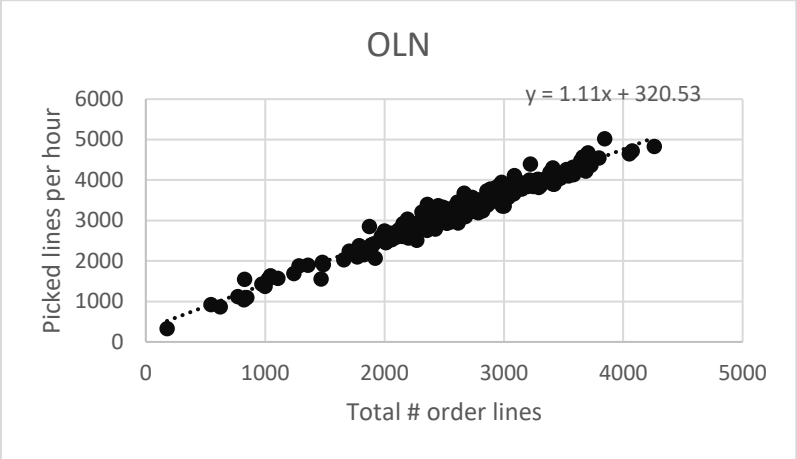


Figure 25: np 2024 productivity (picked lines per staff hr) & total # order lines

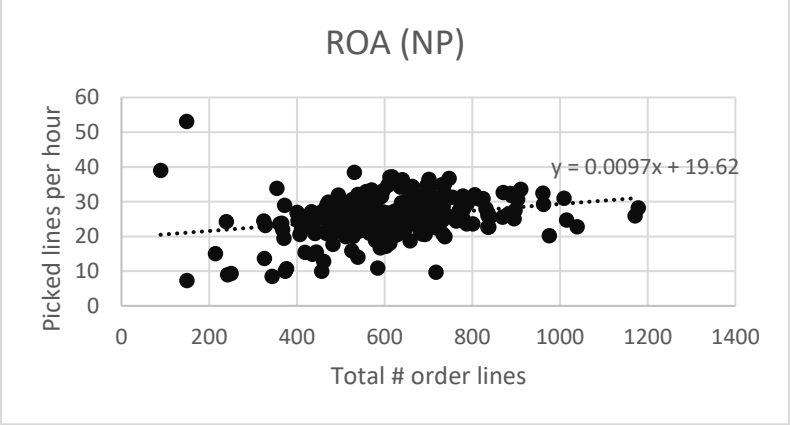


Figure 26: sp 2024 productivity (picked lines per staff hr) & total # order lines

A.3 Quick and dirty problem overview

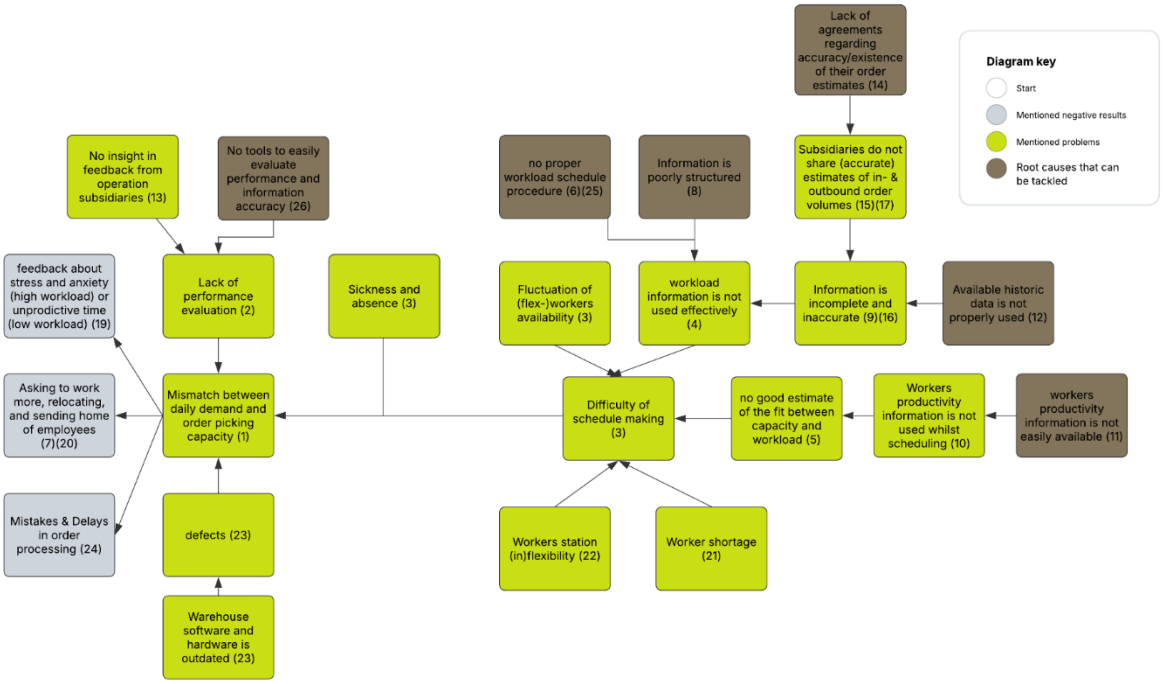


Figure 27: Quick and dirty problem overview

Scheduling and Planning Issues

1. Schedules and actual workload do not match (due to high sick rates, unpredicted technical malfunctions, and a mismatch between prediction and reality of the workload).
2. Evaluation of employee schedules, their productivity and used information accuracy is no fixed part in the scheduling procedure.
3. Making schedules is difficult (flex workers availability fluctuates substantially, order volume information is insufficient, sickness & absence, workers cannot work at all stations (workers inflexibility)).
4. In the scheduling process, the workload information is not a driving factor in determining the schedule.
5. There is no good estimate of the fit between the capacity and workload until the day itself.
6. Scheduling workload in advance and forecasting workload is not done.
7. Employees have to be re-located, send home, asked whether they can step in, or work overtime as the result of mismatches between the workload and capacity.

Information and Data Issues

8. Information regarding order volumes is not/poorly structured causing mistakes or missing valuable information whilst making schedules.
9. Information regarding order volumes is incomplete and inaccurate.
10. Workers' productivity information is not used whilst scheduling.
11. Workers' productivity information is not available in an easy-to-use way.
12. Available historic data is not properly used to predict order volumes and workload.
13. No insight in feedback from operation subsidiaries regarding operations quality.
14. Lack of agreements with operation subsidiaries regarding accuracy of their order volumes.
15. Subsidiaries do not share estimates of inbound and outbound order volumes.
16. There is no accurate workload data.
17. Quality of operation subsidiaries data is insufficient.

Employee and Workload Issues

18. Workers' productivity is not fixed but varies per employee.
19. Warehouse employees share dissatisfaction with workload when it exceeds their capacity and may perceive anxiety and stress due to high work pressure.
20. Overworking (after 18:00).
21. Worker shortage.
22. Limited workers' station flexibility.

Operational and System Issues

23. The warehouse software and hardware are outdated (resulting in defects).
24. Issues cause mistakes and delays in order processing
25. There is no structured set of rules to ensure in-time processing is finished before orders are to be send away
26. No tools to easily evaluate performance and information accuracy
27. Warehouse employees have no insight in ETF (Expected Time Finished) on the day itself. *Split the dashboard in a way that also the information on the day itself is used to provide ETF with current number of employees.*
28. No insight in the productivity levels compared to earlier years.

A.4 Current forecasting performance

$$Error = Demand - Forecast \quad , \quad Percentage\ Error = \left|\frac{Error}{Demand}\right|$$

Date	Wk	Forecast	Demand	Error	Percentage Error
30/12/2024	1	2300	1234	-1066	86.4%
31/12/2024	1	733	482	-251	52.1%
02/01/2025	1	1300	1339	39	2.9%
03/01/2025	1	1300	1225	-75	6.1%
06/01/2025	2	2500	1866	-634	34.0%
07/01/2025	2	1600	2862	1262	44.1%
08/01/2025	2	3300	4198	898	21.4%
09/01/2025	2	2600	1557	-1043	67.0%
10/01/2025	2	3100	3442	342	9.9%
13/01/2025	3	1700	4267	2567	60.2%
14/01/2025	3	1350	1907	557	29.2%
15/01/2025	3	2550	3439	889	25.9%
16/01/2025	3	3000	2925	-75	2.6%
17/01/2025	3	4000	4665	665	14.3%
20/01/2025	4	4200	3686	-514	13.9%
21/01/2025	4	2800	3219	419	13.0%
22/01/2025	4	5400	4411	-989	22.4%
23/01/2025	4	3000	3262	262	8.0%
24/01/2025	4	3400	3779	379	10.0%
27/01/2025	5	3250	4134	884	21.4%
28/01/2025	5	2500	3040	540	17.8%
29/01/2025	5	4500	4314	-186	4.3%
30/01/2025	5	3100	3043	-57	1.9%
31/01/2025	5	3600	3843	243	6.3%
03/02/2025	6	6240	5807	-433	7.5%
04/02/2025	6	5200	3192	-2008	62.9%
05/02/2025	6	3900	4676	776	16.6%
06/02/2025	6	3350	3693	343	9.3%
07/02/2025	6	4950	3900	-1050	26.9%
10/02/2025	7	2600	4780	2180	45.6%
11/02/2025	7	4100	2389	-1711	71.6%
12/02/2025	7	3200	4107	907	22.1%
13/02/2025	7	4400	3061	-1339	43.7%
14/02/2025	7	6200	3232	-2968	91.8%
17/02/2025	8	5662	3700	-1962	53.0%
18/02/2025	8	4310	2220	-2090	94.1%
19/02/2025	8	3290	3860	570	14.8%
20/02/2025	8	4706	3118	-1588	50.9%
21/02/2025	8	4022	4699	677	14.4%
		133213	***128573***	***-4640***	**30.8%**

Figure 28: wk 1-8 2025 forecasting error

$$MAPE = \frac{1}{n} * \sum_{i=1}^{n} Percentage\ Error_i \quad , \quad Bias = \frac{\sum_{i=1}^{n} Error_i}{\sum_{i=1}^{n} Demand_i}$$

Measure	Value
MAPE	*30.8%*
Bias	-3.5%

Figure 29: MAPE and Bias of current forecast method

A.5 Decomposition of demand 2023-2024

The demand data from 2023 and 2024 was retrieved and aggregated to daily level.

The original data consisted of the first 4 columns like in the provided subset of the data:

date	month	day	Sum of Demand	without trend	without seasonality	without level
2022-01-03	1	2	1129	1129	1034.286602	-797.1882831
2022-01-04	1	3	1406	1403.999142	2068.198711	236.7238253
2022-01-05	1	4	1043	1038.998284	1042.957205	-788.5176806
2022-01-06	1	5	1198	1191.997426	1303.958935	-527.5159508
2022-01-07	1	6	1031	1022.996568	941.8807649	-889.5941204
2022-01-10	1	2	1526	1515.99571	1388.816698	-442.6581874
2022-01-11	1	3	1188	1175.994852	1732.330857	-99.14402846
2022-01-12	1	4	1248	1233.993994	1238.695912	-592.7789736
2022-01-13	1	5	1401	1384.993136	1515.082277	-316.3926086
2022-01-14	1	6	1221	1202.992278	1107.604192	-723.8706934
2022-01-17	1	2	1568	1547.99142	1418.128243	-413.3466424

Table 11: Subset of historic demand

The average demand from 2023 and 2024 were compared to determine the total trend. This value was divided by the total number of periods and the daily demand was subtracted by this value times the value of the period the instance was. Then weekly and monthly seasonality factors were determined by dividing the average of that period by the overall average demand. And then seasonality was removed by dividing by the determined factor. Finally, the remaining average was subtracted from the instances to account for the level. The remaining random component turned out to be 33.10% of the total fluctuation.

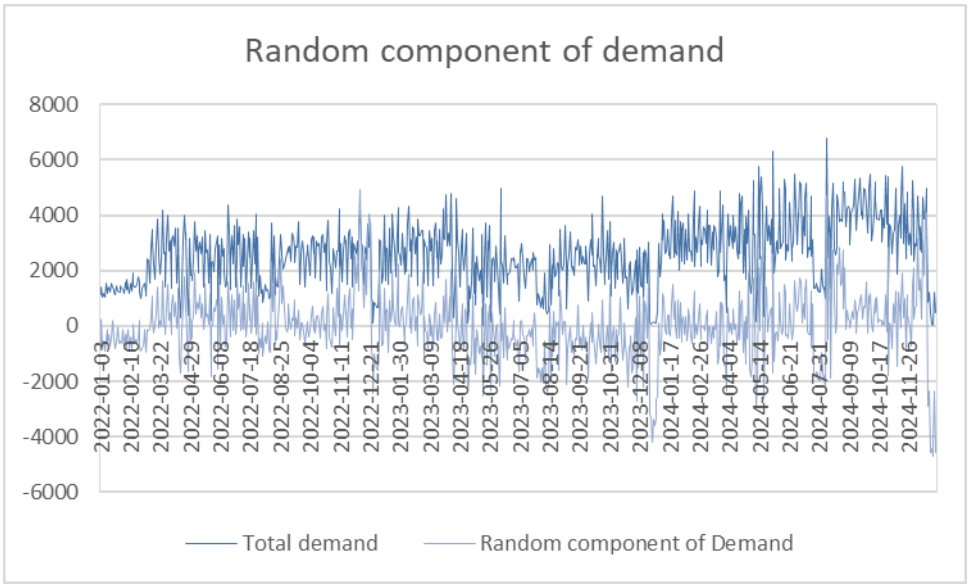


Figure 30: Random component of demand

A.6 Internal processing rules

1. **Rule**: Order customer I: Day 1 the order is placed, Day 1&2 it must be processed by 23:59, Day 3 it is picked up at 8:00.
2. **Rule**: customer K. Day 1 the order is placed, Day 1&2 it must be processed by 23:59, Day 3 loaded by 15:00. (These orders must stay together in the system but can be brought forward for planning purposes.)
3. **Rule**: customer L. At least 50% of the order info comes on Monday and the rest on Wednesday for the Monday shipment. Must be finished by Friday. Must be spread over the days.
4. **Rule**: Customer O comes in on Monday and must be finished by Thursday. Friday departure date. Urgent orders can still be processed on Friday morning.
5. **Rule**: Customer P comes in on Wednesday and must be finished by Monday along with the fast orders. Tuesday departure.
    - General rule for P: set a maximum number to be processed per day to avoid overwhelming the expedition. Mainly for air. Must be evenly distributed over the available days.
6. **Rule**: Other customers: Ordered before 5:30 PM must be processed the same day. Applies to C, E, F, etc.
7. **Rule**: Spares and transfers must also be processed the same day if ordered before 5:30 PM.
8. **Rule**: D must be registered by Tuesday 3:00 PM. What is in the system at that moment must be finished by Monday. Orders usually start to come in on Friday.
9. **Rule**: Customer C Wednesday and Friday shipments as in rule 6.
10. **Rule**: Urgent shipments (via SP6) go through the service desk, are always small orders and therefore cannot be planned, but do not require much work. (This is negligible for planning. Optionally check data to see if it is a significant amount.)

Summary: Everything on the same day except for one large customer (multiple subsidiaries), for which there are 3 days to pick everything. (however, they should be picked the day before they are put on transport)

A.7 Data collection, cleaning, and transformation of customer demand dataset

All details of the transactions within the warehouse are stored in the data warehouse. However, this is done in different datasets, meaning that several datasets had to be collected and pre-processed in order to acquire a dataset that could be used for the analysis in this report. The data cleaning is done on three levels as suggested by Jafari.

1. Clean up the table
2. Unpack, restructure, and reformulate the table
3. Evaluate and correct the values

The overall approach is visualized in Figure 22 and consists of first importing and cleaning part of the shipment's dataset for the period from 2022 to week 13 of 2025. Then the same is done for the OBOD dataset. Finally, these two pre-processed datasets are merged and visualized in a way that demand is disaggregated per customer and summed to daily totals.

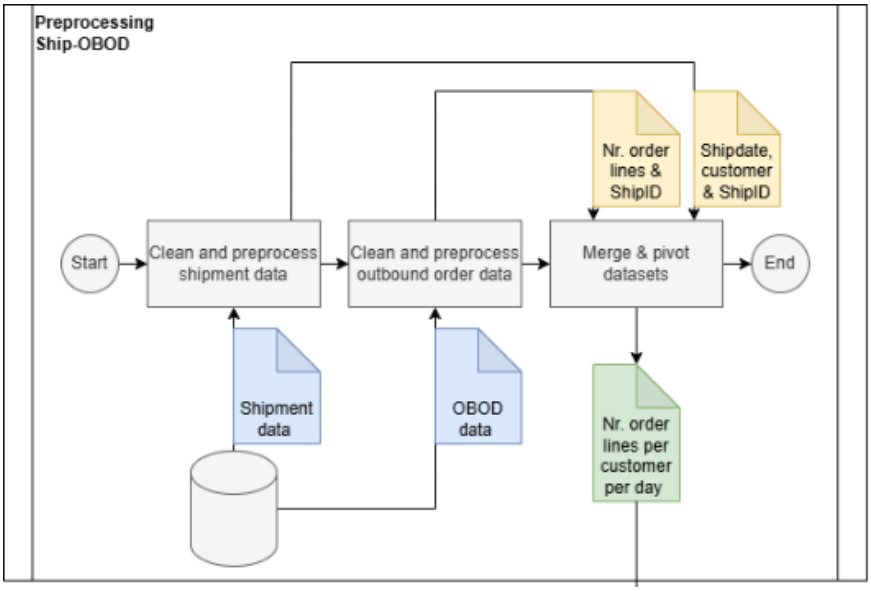


Figure 31: Preprocessing steps to acquire clean dataset with demand per customer

Used dataset 1 – Shipments dataset

The first dataset to use is one with information on the shipments. This dataset includes the "client name", "shipment ID", "date of shipment", and the "original date of shipment" (which is the same as the "date of shipment" if it was not changed) and some more columns that could be used to identify the different companies, as "clients" consists of names that are aggregated to a higher level than desired. This dataset is used to identify per shipment ID the customer (on desired aggregation level), and the shipment date.

Level 1: Clean up the table

The dataset was already in level one as they are conform the standards of a level 1 cleaned dataset: standard data structure, codable and intuitive column titles and each row has a unique identifier (Jafari, 2022).

Level 2: Unpack, restructure, and reformulate the table

In level two we ensure the dataset is in desired format so that the analytics can be done. The main challenge in level two was identifying the different customers. In the shipments dataset the "client" is

specified. However, two customers had to be further disaggregated to arrive at the desired aggregation level.

Through the “CCCODE3” column a distinction could be made between customers K and L as company K had a unique identifier in this column. Through the ship class a distinction could be made between customers N, O, and P as they have different ship classes. Through a unique identifier in the account column of the shipment dataset, companies D and E could be disaggregated. Company M consists of two distinct types of customer demand but was aggregated to one as it is considered to be one customer in reality.

Level 3: Evaluate and correct the values

Through assessment of the recorded dates in the shipment dataset six dates were identified that were in different format, leading to problems with the analysis. The format of these dates was manually corrected as the correct date could still be derived from the wrongly formatted dates.

Next to that, in the dataset 6627 shipments were set during weekends (sat/sun) from the total 547904. Outbound/Expedition sometimes has to (due to max. nr. shipments per day) put a shipment on another day even though they send it on the original day (DTEXWORKSORG). There were 931 instances in which the original date was a weekday, but the stored date was not. These were set to the original date as they are caused by this action of outbound and were actually send for transport on the original date. This was 14.05% of all instances that are during weekends. The remaining 5696 instances are internet orders and are processed on Mondays. So, they were set to the Monday after the weekend they are located.

Used dataset 2 – Outbound orders dataset

The second dataset contains for each shipment ID a row of data for each order line. From this dataset we aim to identify the total number of order lines. We collect the shipment id, order line, customer name and date of creation.

Level 1: Clean up the table

The dataset was conform the standards of a level 1 cleaned dataset: standard data structure, codable and intuitive column titles and each row has a unique identifier (Jafari, 2022).

Level 2: Unpack, restructure, and reformulate the table

To ensure the analytics can be done we need to have the total number of order lines per shipment ID. We restructure the dataset by keeping per shipment ID only the highest value of the stored order line, as this is equal to the total number of order lines as for every shipment the order lines start counting upwards from 1.

Level 3: Evaluate and correct the values

No anomalies were found in the data.

For the final dataset, the two datasets are merged on the shipment ID such that per shipment we know the customer, date of shipment, and number of order lines.

To ensure no data was lost in this merge, the number of shipments was used as the base dataset and left join of the number of order lines resulted in zero shipments without a number of order lines, so all shipments in the shipment dataset no included the total number of order lines.

Although all shipments in the Shipment dataset were assigned the number of order lines, 72,678 shipment IDs from the outbound orders dataset could not be merged as their shipment ID was not present in the shipment dataset. This selection was further analysed to determine the reason for this mismatch. It was determined that due to the decision to reuse shipment IDs for customer C in 2023 these shipment IDs could not be stored in the shipment's dataset in the data warehouse as it was seen as duplicates. Luckily, customer C orders are always send on the same day as they are created, unless they are created in weekends, then they are processed on the next Monday. Therefore, the creation date from the outbound orders dataset could be used as the departure date and the instances were this specific issue prevented the merge could still be added. This explained and managed 61,034 of the instances. The remaining 11,644 shipments that could not be merged were further analysed and it was found that:

1. 1 shipment had been deliberately removed as it was for a customer that is no longer present.
2. 10,363 shipments in the outbound order dataset had departure dates before 2022-01-01 and were therefore excluded from the shipment dataset.
3. 1,279 shipments in the outbound order dataset had departure dates after 2025-03-31 and were therefore excluded from the shipment dataset.
4. 2 shipments were missing in the shipment dataset for unexplainable reason. As the impact of these two shipments is negligible due to their small order size, the decision was made to not further research these two shipments but ignore them.

Having explained and assessed the quality of the merge we create a pivot table with for every date the sum of the number of order lines per customer. The final dataset is over the period 2022 to the last complete week of March 2025 which is ISO week 13.

A.8 Reconstructed method test period results

DAYS AHEAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MWAPE (%)	73	73.4	79.6	79.9	80.3	81.8	81.4	82.5	91.2	84.2	84.8	85.3	86	87.3	86.8	90.9	92	161.9	165	169.1
BIAS (%)	-3.3	-5.5	-10.8	-9.9	-9.5	-9.9	-10.4	-9.8	-9.9	-11.2	-12.6	-12.8	-12.7	-12.9	-13.6	-7.8	-7.3	-0.5	2.4	3.6
MAD	799.1	819.7	997.6	1001.4	1045.7	1077.4	1089.8	1101.2	1528.9	1149.6	1180.4	1222.6	1221.5	1232.2	1264.6	1332.3	1410.6	1475.5	1553.3	1672

Table 12: Customer performance reconstruction on a 7- and 20-day horizon

Reconstruction	*days ahead*	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*	*J*	*K*	*L*	*M*	*N*	*O*	*P*
MAD	7 days MAD	1.8	2.7	257.4	25.4	32	45.7	1.5	4.5	66.2	24	174.9	173.1	80.5	0	80.8	5.4
	20 days MAD	1.8	2.7	257.4	25.4	32	45.7	1.5	6.9	135.8	24	221.6	287.4	80.5	0	80.8	5.4
Bias (%)	7 days bias	-7.6	30.7	-8.5	-11.9	-21.3	-57.9	-86.1	8.2	0.2	31.7	5.4	-8.1	-47.7	-	-50.4	16.2
	20 days bias	-7.6	30.7	-8.5	-11.9	-21.3	-57.9	-86.1	18.2	0.6	31.7	11.5	-9.7	-47.7	-	-50.4	16.2

Table 13: Reconstructed method overall performance

Customer bias (%) Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6	-7.6
B	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7	30.7
C	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5	-8.5
D	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9	-11.9
E	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3	-21.3
F	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9	-57.9
G	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1	-86.1
H	37.9	3.9	3.4	2.8	3.3	3.6	2.5	3.9	3.0	3.3	4.0	3.3	2.8	3.5	3.5	30.9	20.1	82.7	83.7	61.9
I	5.0	4.6	-0.2	-2.4	0.0	-2.2	-3.7	-1.9	-3.8	-3.3	-4.7	-7.6	-7.0	-6.9	-9.4	-11.8	-6.4	9.3	16.8	46.9
J	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7	31.7
K	15.3	7.5	1.6	4.0	2.9	3.6	3.0	3.0	88.5	0.0	-2.4	-4.0	-6.0	-7.3	-7.4	38.2	32.7	27.1	26.2	4.3
L	2.2	-1.3	-13.9	-10.8	-10.3	-11.0	-11.7	-10.5	-38.8	-13.3	-16.3	-15.0	-14.6	-14.5	-15.7	-12.1	-11.0	4.4	10.5	8.9
M	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7	-47.7
N	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan	nan
O	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4	-50.4
P	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2	16.2

Figure 32: Reconstructed method customer bias (%)

Customer MAD Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8	1.8
B	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7	2.7
C	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4	257.4
D	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4	25.4
E	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0
F	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7	45.7
G	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5	1.5
H	9.3	3.6	3.6	3.8	3.7	3.6	3.6	3.6	3.7	3.6	3.6	3.6	3.8	3.6	3.6	6.8	8.7	19.4	19.4	23.1
I	32.9	27.8	56.2	64.3	80.5	95.3	106.4	111.2	128.4	127.1	126.8	139.7	133.3	127.8	143.2	172.6	197.6	240.3	278.1	326.2
J	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0
K	122.1	149.7	172.5	190.2	194.7	197.1	198.2	197.9	320.4	208.4	209.2	217.1	223.7	227.7	233.1	247.2	260.5	276.6	273.6	311.9
L	77.7	81.3	208.2	186.1	209.6	224.2	224.4	231.4	519.2	253.4	283.6	305.1	303.6	316.0	327.6	348.5	386.7	382.0	425.1	453.7
M	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5	80.5
N	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
O	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8	80.8
P	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4	5.4

Figure 33: Reconstructed method customer MAD

Notice that the black line is the highlighted 20 day ahead prediction, and the green line the 1 day ahead prediction. For customers for which only historic data is used for the prediction, these are the same predictions, so in that case only the black line is shown.

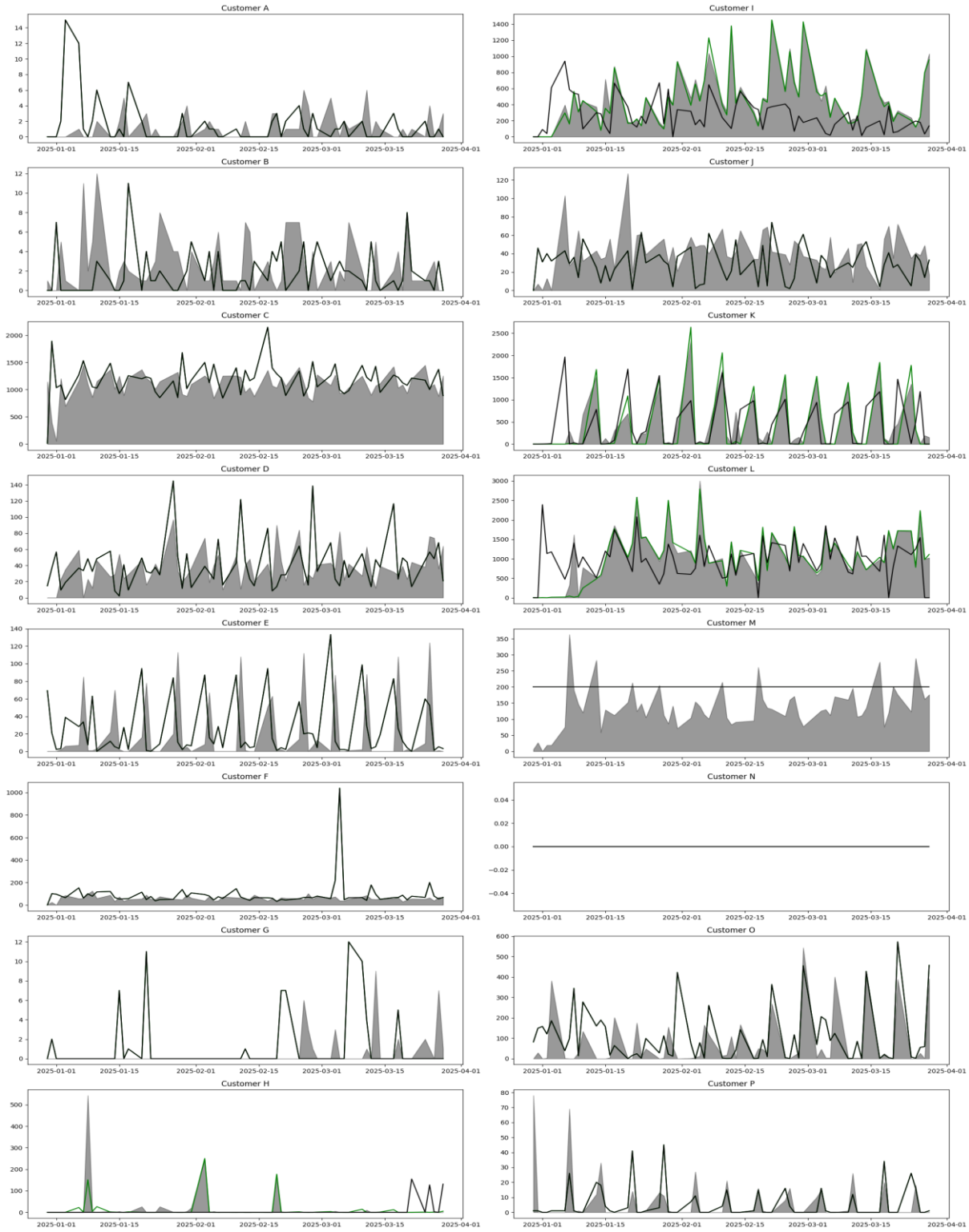


Figure 34: 1 and 20 day ahead predictions for test period reconstructed method

A.9 Seasonality assessment of customer demand

Seasonality is assessed for three potential seasonal patterns for all customers. One level to assess seasonality within the week, and two levels to assess seasonal patterns over the year. Looking at the demand for the weekdays (Figure 26) it turns out that a within-week seasonality seems to be present for most customers. But that this seasonality pattern is very different. For example, customer H shows a peak on Wednesdays, O on Fridays, and P on Tuesdays. Looking at the plots with a weekly seasonal pattern (Figure 27) the results are different. Most customers do not show clear distinctions between weekly season, but instead a more stable line. This seems to be the case for customers A-G, although we cannot be completely certain as some reoccurring peaks do seem to be present, especially around Christmas. Customers H-P seem to have some seasonality over the year, for example during the construction holiday, and around the Christmas days. Looking at the monthly plots (Figure 28) we may reconsider whether A-G are not subject to seasonality over the year. For example, customer D also seems to be subject to lower demand during the construction holiday in months 8 and 9, and customer A seems to have structurally lower demand in the second and third month of the year.

Figure 35: customer demand per weekday

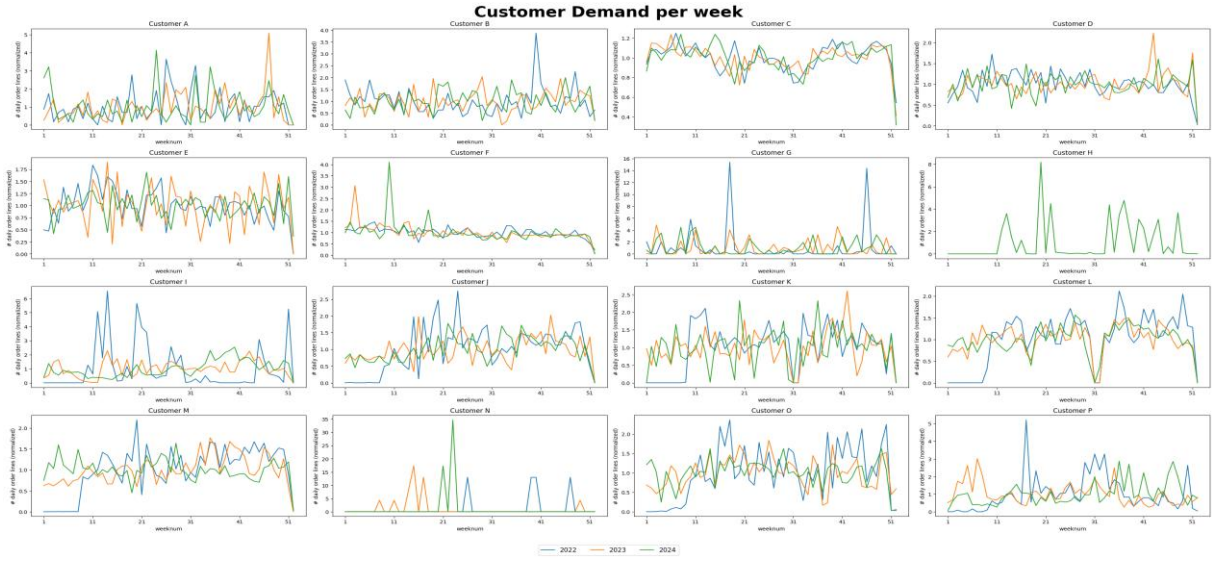


Figure 36: customer demand per week

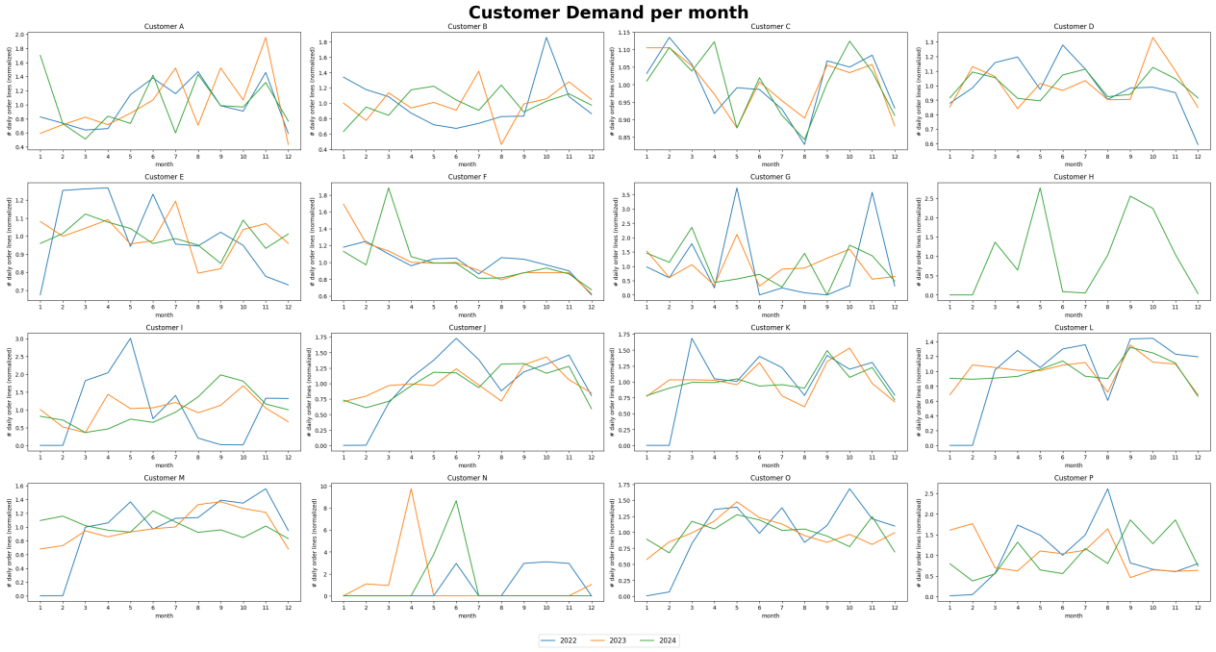


Figure 37: customer demand per month

Whilst analysing seasonality within the weeks and throughout the year, the idea arose that national holidays may affect demand. To get a better understanding of this potential effect on demand, we plot the mean demand during holidays, the day before and after holidays, and the overall mean demand to see if there seems to be a relationship between national holidays and customer demand. For all customers except customer P, it seems that demand during a national holiday is lower than normally. For some customers (like E, F, G, J, K, L, N, O) this decrease in demand seems to be (partially) transferred to the day before or after the holiday. As there are few national holidays within the dataset (less than 20) the results are likely not completely robust. Still the results do tend to suggest that holidays may affect daily customer demand.

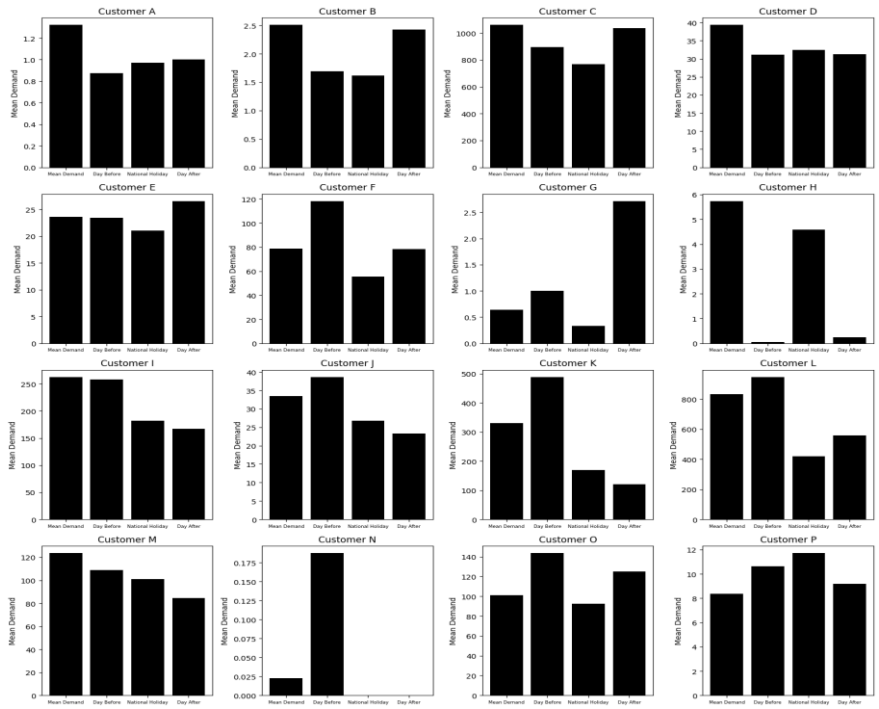


Figure 38: Customer demand on and around national holidays

A.10 STL decomposition of customer demand

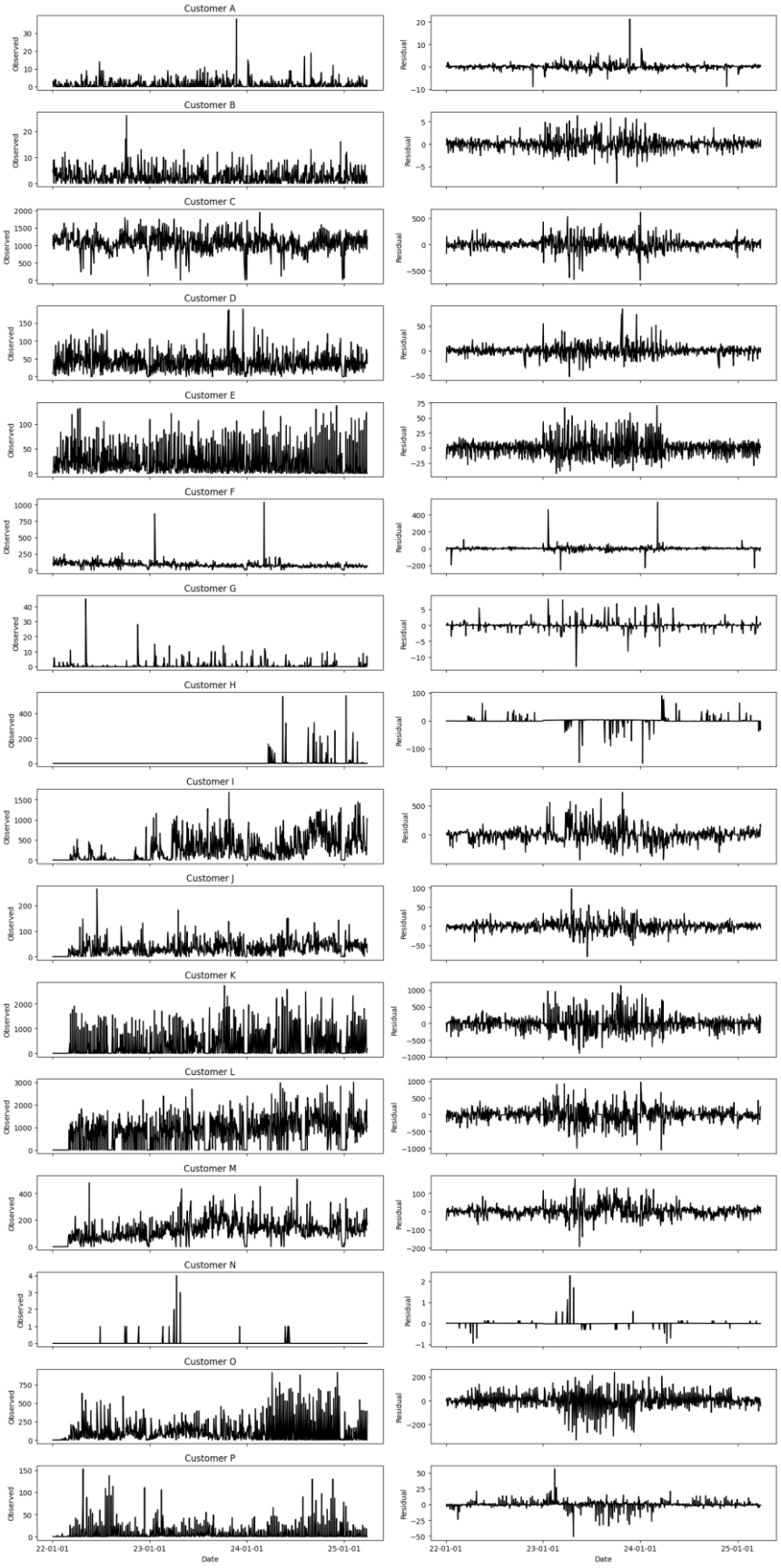


Figure 39: STL decomposition of customer demand

A.11 Basic Prophet model configuration and results

Basic Prophet model configuration and hyperparameter choices

NeuralProphet applies normalization techniques before training the model. The available normalisation options within the NeuralProphet framework are 'off', 'soft', 'soft1', 'minmax', and 'standardize.' Normalizing the data is recommended for deep learning methods (Bhanja & Das, n.d.). As we will eventually include deep layers, we should thus select a normalization technique (i.e. not select "off"). By default, soft data normalisation is used for non-binary data (Triebe et al., 2021). This default method scales the minimum value in the training data to 0.0 and the 95th quantile to 1.0. Soft1 scales from 0.1 to 1 for the 90th quantile. Minmax strictly scales all data to a [0,1] interval making it more sensitive to outliers than previous options. The standardize method zero-centers the training data and divides it by the standard deviation.

The customer demand of TKHL is non-negative and shows frequent zero observations. The 'standardize' method is undesired as it builds upon the assumption that the data has a Gaussian distribution which is unplausible due to a large spike for zero demand (Jamal et al., 2014). Certain customers have outliers in their demand. We prefer the soft option over minmax as the strict [0,1] scaling of the latter makes it more sensitive to these outliers. Soft1 scales from 0.1, so to keep zero demand soft normalisation is preferred instead.

NeuralProphet provides two optimizers. AdamW (Loshchilov & Hutter, 2017) and Stochastic Gradient Descent (SGD). AdamW is mentioned as a reliable default option (Triebe, 2024) and the SGD optimizer requires more fine-tuning of hyperparameters making the AdamW optimizer the preferred choice.

Users can select any loss function that matches the PyTorch loss function format. The default loss function is smooth L1-loss (or Huber loss) which is either the MAD or MSE based on a threshold β that is set to 1. Huber loss uses the MSE if the error is less than the threshold, and the MAE otherwise. This makes smooth L1-Loss less sensitive to outliers than pure MSE (Girshick, 2015). The MSE is generally used as the loss function in regression problems. On regression problems with many outliers, we may however want the loss function to be more robust. The default loss function does this by using the MAD when the error is at least 1. The MAD is more appropriate with outliers as it does not heavily punish errors with outliers, which the MSE does due to its quadratic term. The default threshold of 1 is however very large considering the normalization technique that results in 95% of data being scaled in between 0 and 1. Therefore we do not use smooth L1-loss, but default L1-loss (MAD).

With NeuralProphet the learning rate is increased and decreased exponentially and using the log10-mean of three runs the learning rate is selected according to the learning rate range test (Smith, 2015). The number of observations in the training set for most of the customers is 650 business days. The batch size (B) and trainings epochs (N_{epoch}) are determined as:

$$B^* = 2^{2+\lfloor \log(650) \rfloor} = 2^4 \approx 16$$

$$B = Min\Big(650, Max\big(16, Min(256, B^*)\big)\Big) = 16$$

$$N^*_{epoch} = \frac{1000 * 2^{\frac{5}{2}*\log(650)}}{650} \approx 201.4$$

$$N_{epoch} = \min\big(500, \max\big(50, \lfloor N^*_{epoch} \rfloor\big)\big) = 201$$

The NeuralProphet model could provide negative predictions. Harvey & Ito (2020) suggest that shifting and censoring can deal with such negative values. A more straightforward method similar to this approach is to clip the negative predictions. This implies the replacement of all negative predictions by zero. We use this method to deal with potential negative predictions.

Running the model with the default settings once for all 16 customers using the CPU of a ThinkPad P15v takes approximately 7 minutes. As the time series cross validation on the test set requires this procedure to be done 85 times, and later model extensions will probably increase computation times, we decide to increase the batch size to 64, which follows after 16, to reduce the training time.

Basic Prophet model comparison additive & multiplicative seasonality

Model settings:

model=NeuralProphet(n_forecasts=20,yearly_seasonality=True[2],weekly_seasonality=True,daily_seasonality=False,normalize="soft",optimizer="AdamW",loss_func="L1",batch_size=64, epochs=201)

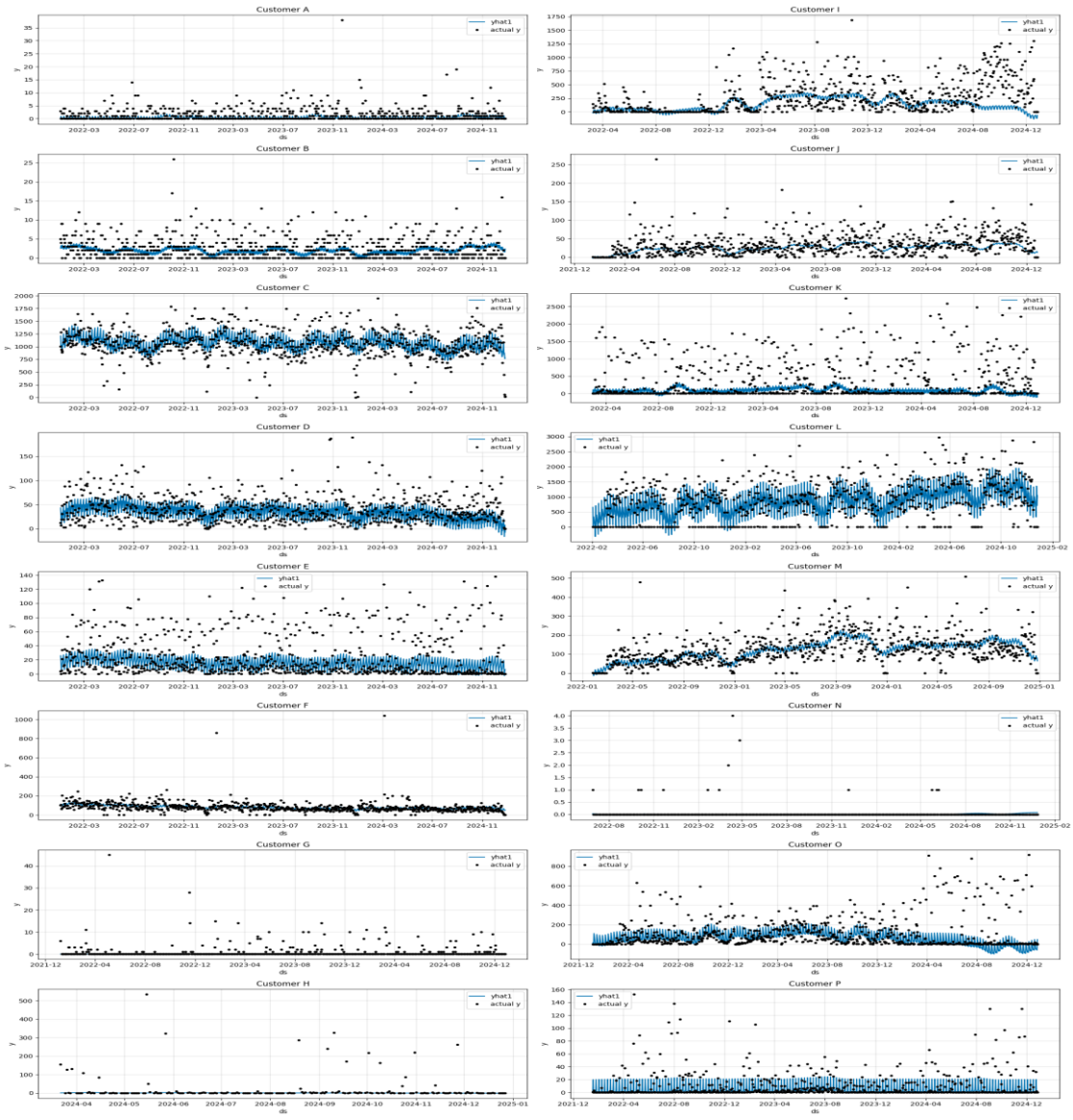


Figure 40: fitted trend and yearly and weekly additive seasonality Prophet model – batch size = 64

[2] As mentioned in the report yearly seasonality is turned of for customer H due to lack of data

Same as before, but “multiplicative” seasonality_mode for customers I, L, O, and P

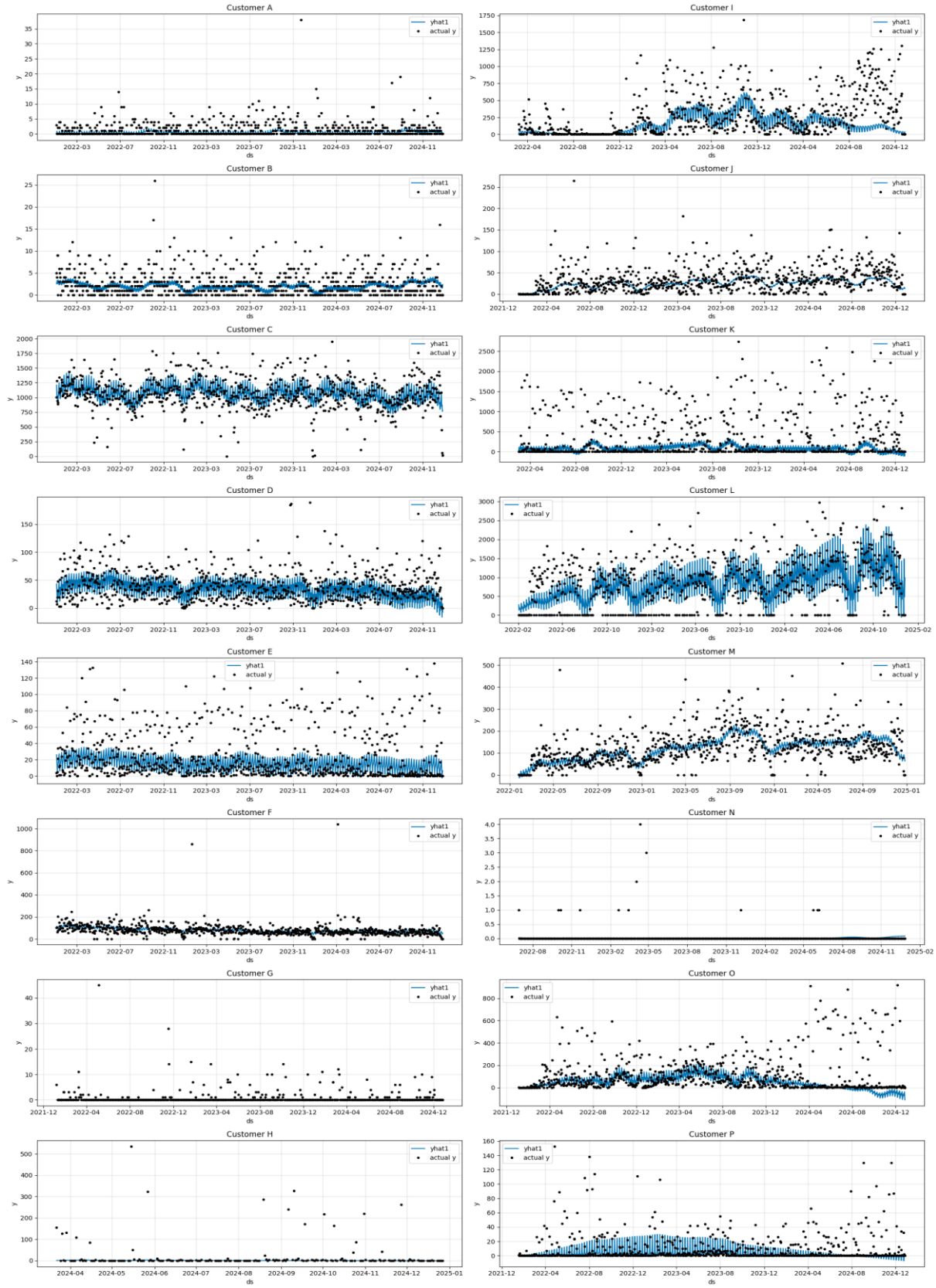


Figure 41: : fitted trend and yearly and weekly multiplicative (for I,L,O,P) seasonality Prophet model – batch size = 64

Basic Prophet test period results

	days ahead	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
MAD (% improvement against reconstruction)	7 days	33.3	11.1	30.5	27.2	41.3	57.3	53.3	-444.4	-401.7	29.6	-108.9	-154.1	37.5	-	39.0	7.4
	20 days	33.3	7.4	25.5	25.6	39.4	55.1	53.3	-240.6	-156.0	28.8	-73.7	-65.0	38.1	-	37.0	5.6
Bias (%)	7 days	75.5	44	2.8	1.2	53.2	0.5	68.6	50.3	-1.2	5	46.9	2	9.1	-	37.4	8.1
	20 days	75.1	42.4	3.6	-0.4	50.3	-3.6	72.1	54.1	-4.6	4.8	41	-4.4	9	-	34.7	6.5

Table 14: Customer A-P performance trend & seasonality model

DAYS AHEAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MWAPE (%)	116.6	123.4	121.4	131.7	139.6	127	151.3	151.1	158	153.5	131.8	188.3	150.7	135	147.1	145.8	147.6	151.9	155	205.9
BIAS (%)	9.3	8.7	7.5	8.5	7.7	8.7	7.5	5.9	6.8	5.9	7.5	5.8	3.8	3.7	3.3	4.1	1.9	-1.3	-0.2	-0.7
MAD	1413.8	1553	1500	1581.1	1500.5	1521.1	1594.2	1625	1682	1593.8	1592.5	1703.3	1681.5	1667.2	1575.8	1666.3	1668.1	1732.6	1671.1	1648

Table 15: Basic Prophet overall performance

Customer bias (%) Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	74.0	73.8	76.8	77.0	76.8	75.3	74.8	78.0	79.2	76.5	71.7	62.1	76.3	81.4	79.1	76.0	63.2	72.7	79.4	77.0
B	40.6	49.4	42.9	34.8	48.6	41.8	49.7	42.5	42.4	37.1	34.1	47.1	45.5	44.6	41.3	40.7	49.3	46.1	37.3	31.2
C	2.9	2.8	1.6	2.8	4.3	2.6	2.5	1.0	4.0	6.1	6.7	5.1	1.5	3.1	5.8	7.0	3.6	1.5	3.1	5.0
D	1.6	2.5	4.4	1.0	-1.8	-1.0	1.9	2.0	0.1	-3.0	-6.5	0.2	0.3	-2.0	0.7	-5.7	5.4	-2.4	-3.4	-1.6
E	51.2	52.4	56.5	57.7	54.8	49.7	50.2	53.3	57.8	53.8	51.0	48.5	53.0	46.9	43.5	44.0	43.4	49.5	45.6	42.8
F	0.8	2.6	5.1	2.5	-4.4	-3.0	-0.1	-1.0	-2.4	-4.9	-4.2	-6.9	-4.2	-9.7	-8.1	-2.7	-5.1	-5.8	-11.3	-8.4
G	51.8	74.3	73.5	81.7	75.4	57.5	66.1	62.9	87.3	72.2	76.1	66.3	71.0	85.2	69.8	75.2	64.8	71.6	87.8	71.2
H	58.5	77.7	39.8	43.8	43.8	32.6	56.2	38.9	42.9	54.3	48.8	71.3	67.2	55.9	54.0	48.7	70.9	66.8	55.7	53.9
I	-0.6	-0.2	0.3	-6.5	2.7	-2.1	-2.3	0.2	-9.8	-2.4	-11.5	-4.7	-2.1	-13.8	-4.9	-15.0	-9.6	-3.4	-3.4	-3.6
J	5.1	1.6	4.2	6.1	6.0	6.0	6.0	6.1	5.9	5.7	5.5	5.3	5.3	4.9	4.5	4.1	3.9	3.7	3.3	2.9
K	45.9	42.5	46.1	50.3	48.3	51.7	43.8	47.1	50.3	46.7	57.5	42.0	45.0	45.6	38.2	45.2	30.5	17.5	11.4	15.3
L	5.1	4.2	0.0	3.5	-3.2	3.1	1.6	-4.5	-1.5	-7.9	-2.1	-2.9	-9.6	-6.2	-11.3	-7.7	-7.6	-15.3	-11.1	-15.5
M	8.5	8.6	10.7	10.5	10.2	7.6	7.4	9.5	9.4	9.2	6.6	6.7	9.3	9.7	9.7	7.5	7.7	10.1	10.4	10.5
N	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
O	38.2	36.0	34.4	39.3	39.9	35.2	38.7	43.2	42.8	37.8	32.5	17.2	35.4	36.5	34.8	37.6	13.5	31.9	37.2	31.4
P	8.9	-23.9	26.6	20.1	10.7	11.0	3.5	23.5	19.5	6.7	9.1	-2.4	15.0	15.1	-0.8	-0.5	-15.8	3.1	11.0	-11.3

Table 16: Customer bias (%) basic Prophet for all days in advance

Customer MAD Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.3	1.3	1.2	1.2	1.3	1.3	1.4	1.3	1.3	1.2
B	2.3	2.1	2.4	2.5	2.4	2.4	2.4	2.5	2.5	2.6	2.8	2.5	2.5	2.5	2.7	2.5	2.6	2.6	2.4	2.9
C	163.9	172.3	194.5	181.1	174.1	179.2	187.8	212.4	211.9	180.3	155.3	226.2	198.0	190.0	186.1	195.8	206.6	231.1	212.8	175.3
D	17.7	18.3	18.2	17.8	19.7	18.8	18.9	18.1	17.9	19.5	19.7	20.7	19.1	19.5	17.8	18.9	18.6	19.8	18.0	20.2
E	18.4	19.4	19.3	18.8	18.2	18.4	19.4	19.7	18.8	18.5	19.3	19.9	19.4	20.4	19.7	21.2	21.6	20.5	20.7	17.4
F	17.3	18.8	16.7	19.1	22.5	21.8	20.0	19.7	18.9	21.8	20.9	20.9	18.3	25.0	21.7	19.2	20.7	21.7	23.8	21.5
G	0.8	0.7	0.7	0.6	0.7	0.8	0.7	0.7	0.6	0.7	0.7	0.7	0.7	0.6	0.7	0.7	0.7	0.7	0.6	0.7
H	23.4	20.2	26.6	25.9	25.8	27.8	22.0	26.7	26.1	15.1	24.9	21.2	22.0	24.0	23.4	25.1	21.5	21.4	23.3	24.2
I	328.2	347.2	311.8	378.9	312.8	314.5	331.4	300.4	393.7	354.0	387.2	363.7	353.1	407.7	341.9	419.9	348.2	318.3	314.4	326.8
J	16.7	17.8	17.2	16.5	16.4	16.9	16.7	17.2	16.8	16.8	16.9	16.9	17.1	17.0	17.1	17.5	17.5	17.5	17.5	17.7
K	326.8	410.4	351.2	378.7	341.9	350.7	397.3	364.9	396.2	356.8	361.4	401.6	357.8	363.5	370.9	372.2	435.2	444.9	473.4	445.2
L	397.2	416.5	437.4	432.7	453.1	466.1	475.5	533.2	469.8	494.7	483.9	488.3	568.3	491.1	473.6	471.6	457.5	529.3	462.5	484.5
M	49.6	51.4	47.9	50.2	50.2	54.4	48.6	53.1	52.7	51.3	48.0	52.4	50.4	51.5	49.9	47.2	45.6	46.1	47.1	47.7
N	0.2	0.2	0.1	0.2	0.1	0.2	0.1	0.1	0.2	0.1	0.2	0.2	0.1	0.0	0.1	0.0	0.1	0.1	0.0	0.1
O	45.4	49.7	50.7	52.2	56.2	43.5	47.4	50.0	50.1	55.3	45.4	61.6	48.7	48.4	43.5	48.1	64.6	52.4	48.5	56.0
P	4.6	6.7	4.3	4.8	5.3	4.4	5.0	5.0	4.6	5.1	4.5	5.1	4.9	4.6	5.5	5.1	5.7	4.9	4.6	6.5

Table 17: Customer MAD Basic Prophet for all days in advance

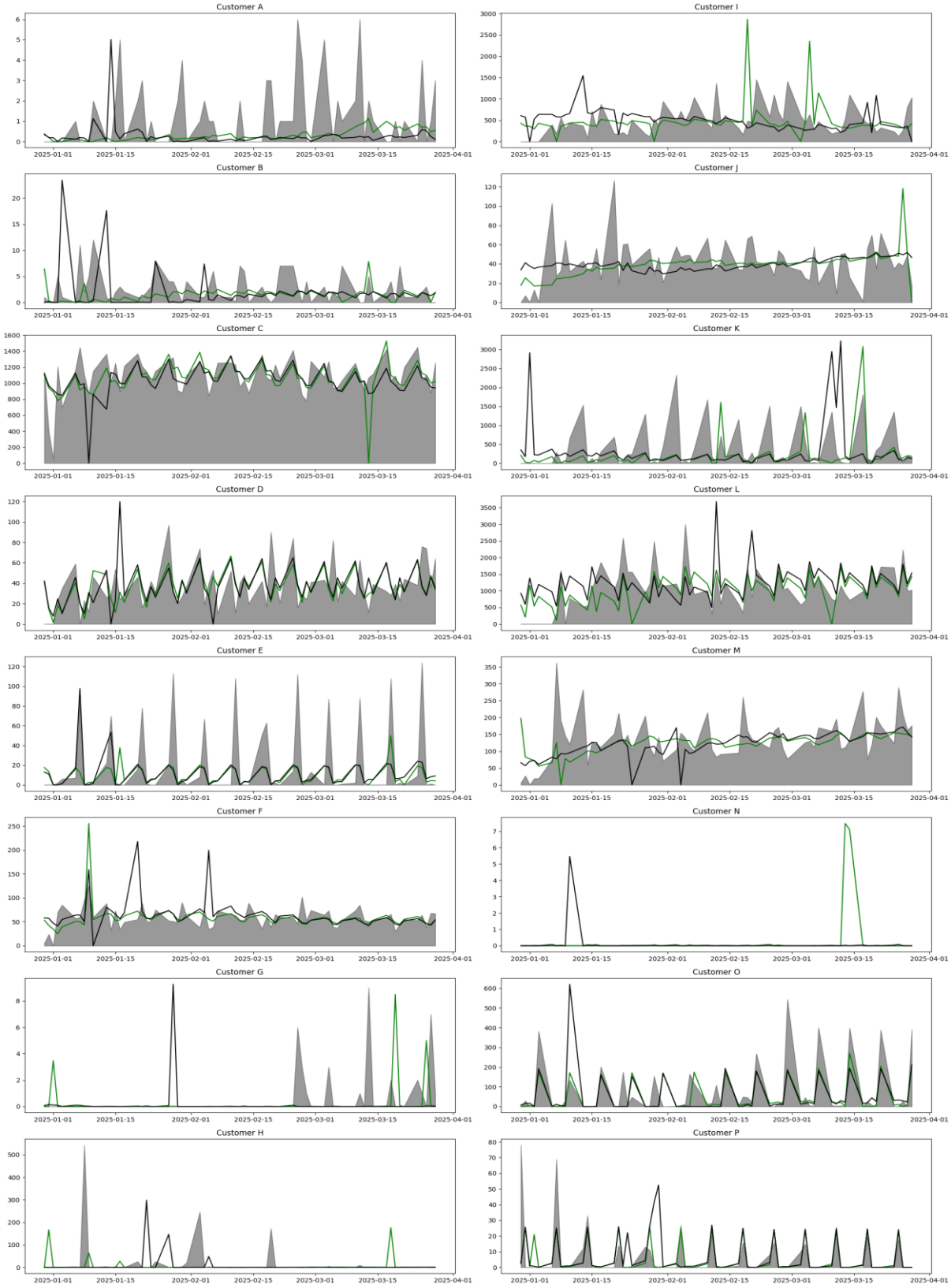


Figure 42: 1 (green) and 20 (black) day ahead predictions for test period Basic Prophet

A.12 Results Prophet with holidays

Results first version – original batch size and learning rate

Model settings:

model=NeuralProphet(n_forecasts=20,yearly_seasonality=True[3],weekly_seasonality=True,daily_seasonality=False,normalize="soft",optimizer="AdamW",loss_func="L1",batch_size=64, epochs=201)

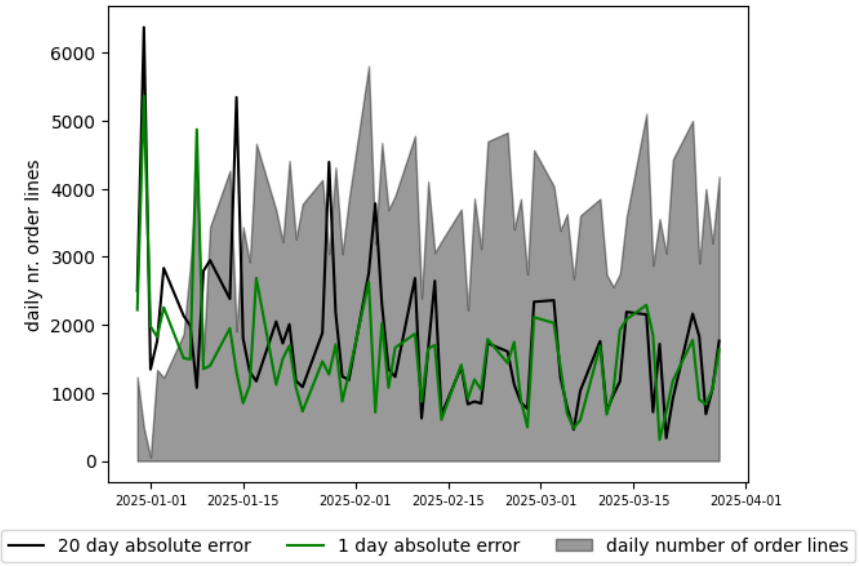


Figure 44: Absolute error Prophet with Holidays 1st results

days ahead	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MWAPE (%)	117.0	116.2	89.0	89.8	85.8	92.8	87.1	104.4	97.7	106.3	144.2	97.5	171.5	98.9	96.7	95.2	95.1	94.9	96.5	113.3
bias (%)	4.4	3.9	6.1	4.4	5.8	4.7	4.0	5.0	3.2	2.7	3.2	1.1	2.6	1.2	-0.1	1.2	-0.3	0.0	-2.5	-4.1
MAD	1506.7	1545.2	1493.7	1553.4	1435.7	1551.9	1575.1	1564.9	1553.1	1588.6	1611.8	1666.6	1612.2	1627.4	1622.0	1636.6	1643.3	1657.9	1684.3	1766.6

Figure 43: Performance metrics Prophet with Holidays 1st results

[3] As mentioned in the report yearly seasonality is turned of for customer H due to lack of data

Batch size and learning rate exploration

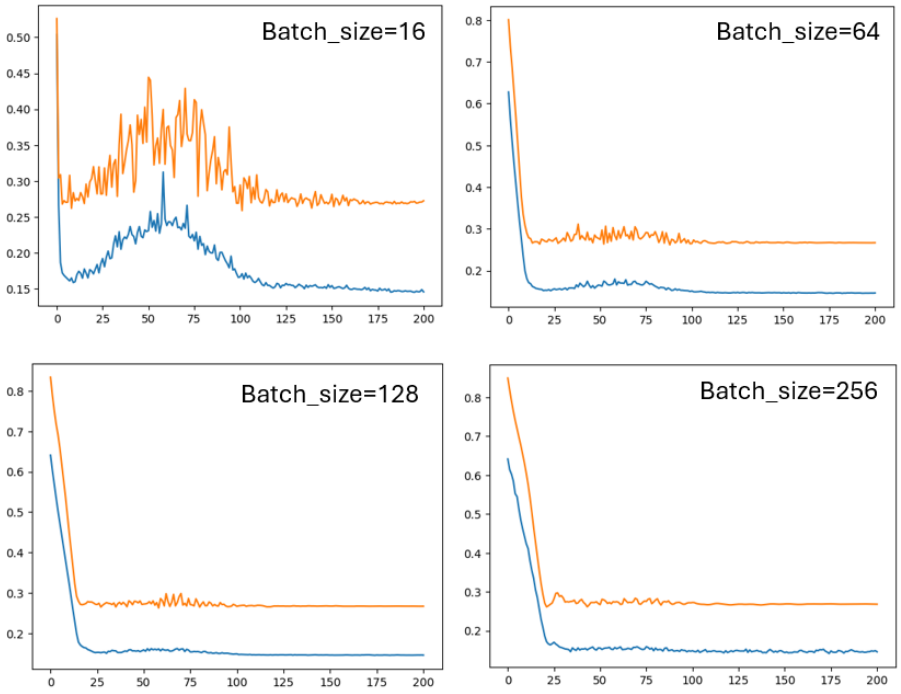


Figure 45: Loss of training (blue) and validation (orange) set per epoch - customer L

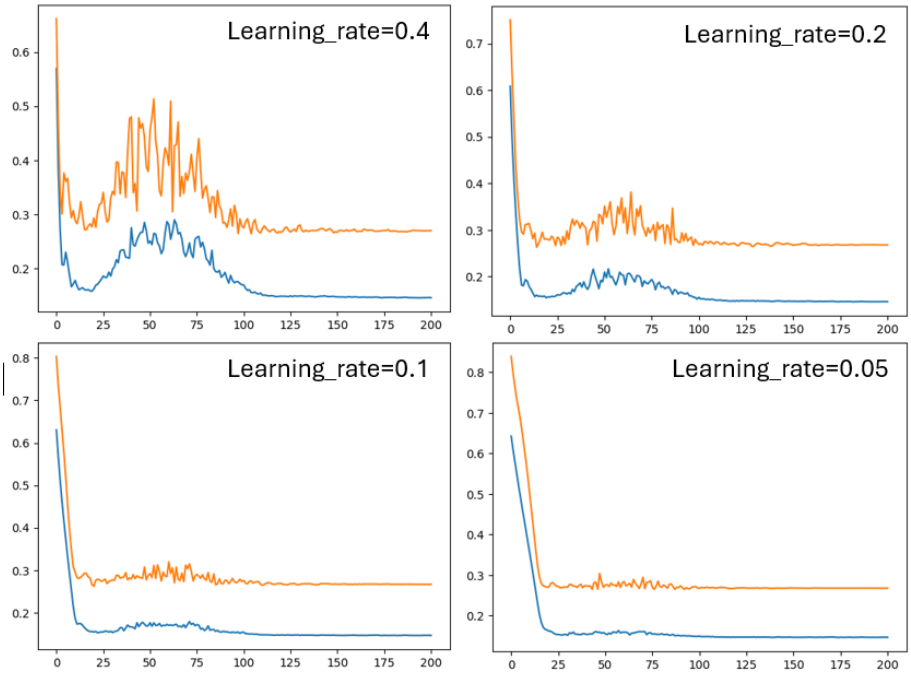


Figure 46: Loss of training (blue) and validation (orange) set per epoch customer L

Prophet with holidays test period results

Trend & Seasonality	days ahead	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
MAD (% improvement)	7 days	33.3	18.5	48.4	33.1	40.3	63.0	60.0	-273.3	-302.6	33.3	-76.2	-125.5	46.5	-	42.9	20.4
	20 days	33.3	14.8	48.3	32.7	40.0	63.5	60.0	-143.5	-108.3	30.4	-39.9	-42.1	47.1	-	41.7	18.5
Bias (%)	7 days	58	51.9	0.3	-4.1	53.6	5	93.5	98.1	10.3	7.2	45.3	-0.7	7.3	-	38.4	12.5
	20 days	61.7	56.5	0.7	-6	53.3	2.4	92.4	98.5	6.8	6.9	41.8	-3.2	6.9	-	37.5	10.8

Table 18: Customer performance holidays model

DAYS AHEAD	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
MWAPE (%)	69.4	72.1	71.5	75	77.9	81.7	81.4	82.6	83.5	82.7	82.7	83.1	82.7	83.7	82.7	83.6	83.3	83.7	83.6	87.3
BIAS (%)	8.8	8.8	8.4	8.3	8.1	7.8	7.7	7.3	7.1	6.8	6.5	6.3	6	5.7	5.7	5	4.9	4.5	4.3	4.2
MAD	1247.6	1275.6	1267.4	1275.5	1287.9	1299.9	1314.1	1301.7	1306.9	1324.6	1337.8	1341.1	1338.6	1344.4	1346.8	1349.5	1364.5	1346.5	1360.7	1355

Table 19: Prophet with holidays overall performance

Customer bias (%) Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	56.8	57.1	57.7	58.1	58.8	58.7	59.1	59.7	59.7	61.3	61.6	62.2	63.5	63.4	64.7	65.3	65.7	66.5	66.0	67.6
B	49.4	49.9	50.9	51.8	52.6	54.2	54.4	55.2	55.7	56.7	58.0	57.7	58.7	59.0	59.6	60.7	60.5	61.4	61.3	62.2
C	0.0	0.6	0.2	0.2	0.4	0.4	0.4	0.4	0.4	0.6	0.7	0.8	1.0	1.0	1.5	0.8	0.8	1.0	0.9	1.1
D	-3.1	-3.5	-3.7	-4.3	-4.2	-4.8	-5.1	-5.4	-5.8	-6.1	-7.0	-7.0	-7.1	-8.2	-7.5	-7.4	-7.4	-7.8	-7.7	-7.7
E	53.9	53.6	53.7	53.9	53.4	53.5	53.5	53.6	52.6	50.6	52.9	53.4	51.8	52.4	52.6	53.5	53.9	54.1	54.2	54.5
F	5.8	5.7	5.2	5.1	4.7	4.4	4.2	3.6	3.2	2.4	1.9	1.9	1.2	1.0	0.5	0.1	-0.0	-0.9	-1.1	-1.3
G	94.5	93.4	93.3	93.2	92.9	93.0	94.5	92.0	94.0	92.2	92.1	92.0	91.7	91.7	91.8	91.6	91.5	91.3	91.0	91.2
H	98.2	98.0	98.1	98.0	98.1	98.2	98.2	98.1	98.3	98.4	98.5	98.6	98.5	98.6	98.7	98.8	99.0	98.9	99.0	99.2
I	12.2	11.8	10.8	10.2	9.6	9.0	8.7	8.0	7.9	7.5	6.9	6.3	5.4	5.0	4.5	3.9	3.3	2.2	1.5	0.9
J	7.3	6.9	7.1	7.0	7.5	7.3	7.2	7.6	7.5	7.5	7.5	7.2	7.3	7.0	6.8	6.4	5.8	5.8	5.4	5.0
K	46.5	46.0	45.5	45.6	44.1	44.8	44.5	43.4	43.3	41.7	42.0	41.4	40.3	40.0	38.7	38.8	38.3	37.3	37.2	36.0
L	0.0	-0.2	-0.4	-0.7	-0.5	-1.4	-1.5	-2.0	-2.6	-2.6	-3.6	-3.8	-4.4	-4.9	-4.8	-5.7	-5.8	-6.4	-6.7	-6.7
M	7.5	7.3	7.4	7.3	7.4	7.1	7.0	6.9	6.8	6.8	6.6	6.5	6.4	6.5	6.6	6.7	6.6	6.8	7.1	7.6
N	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
O	38.6	38.8	39.0	38.2	37.9	38.5	37.8	37.8	37.0	36.7	36.7	36.6	37.7	36.3	36.7	36.4	36.7	38.2	36.9	37.6
P	13.1	12.8	12.4	12.7	12.4	12.0	12.2	11.2	11.5	11.7	10.8	10.8	9.9	10.1	9.7	9.7	9.2	8.5	8.0	8.1

Table 20: Customer bias (%) Prophet with holidays for all days in advance

Customer MAD Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.1
B	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3
C	136.2	128.6	130.9	133.1	131.9	136.9	131.5	133.4	129.6	136.1	141.6	133.1	131.5	129.1	127.5	135.9	134.1	132.4	133.6	137.1
D	17.4	16.9	17.2	16.8	16.7	17.2	16.8	17.1	16.9	17.1	17.3	17.2	17.1	17.7	17.2	17.3	17.0	17.4	16.9	17.2
E	19.2	19.2	19.1	19.0	19.0	19.2	19.0	19.2	19.1	19.7	19.3	19.1	19.6	19.3	19.4	19.6	19.3	19.4	19.1	19.1
F	16.3	16.7	17.1	17.4	16.8	16.8	16.9	16.9	17.0	16.5	16.6	16.2	16.5	16.2	17.1	16.4	16.8	16.6	16.1	17.5
G	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
H	16.8	16.8	16.8	16.8	16.8	16.9	16.9	16.9	16.9	16.8	16.9	16.9	16.9	16.8	16.8	16.9	16.9	16.8	16.8	16.8
I	252.6	265.0	265.0	267.1	272.9	268.8	274.3	273.0	277.0	278.5	281.8	287.7	287.5	295.1	299.2	298.1	304.4	297.6	304.2	308.5
J	15.8	15.9	15.9	16.0	16.1	16.3	16.2	16.3	16.2	16.4	16.9	16.7	17.0	16.9	17.3	17.7	17.2	17.5	17.5	18.1
K	308.5	313.7	300.1	300.4	303.6	311.6	319.4	303.1	306.6	308.0	311.8	315.7	307.3	308.7	307.3	312.3	319.3	313.2	314.5	314.2
L	367.2	384.9	386.5	390.6	398.2	398.8	406.7	408.0	409.8	417.8	416.4	419.2	425.6	423.3	425.2	416.4	421.5	417.7	424.6	410.5
M	43.2	43.2	44.3	44.2	42.1	42.5	42.0	42.7	42.8	43.3	44.2	43.4	42.5	43.7	43.4	41.3	41.2	41.5	40.4	40.4
N	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
O	45.9	46.5	46.3	46.0	45.5	46.7	46.0	46.8	46.4	45.8	46.5	47.4	48.6	49.1	47.9	49.2	48.1	48.0	48.4	47.1
P	4.3	4.3	4.3	4.3	4.3	4.3	4.4	4.3	4.4	4.3	4.4	4.4	4.4	4.4	4.4	4.3	4.4	4.4	4.4	4.5

Table 21: Customer MAD Prophet with holidays for all days in advance

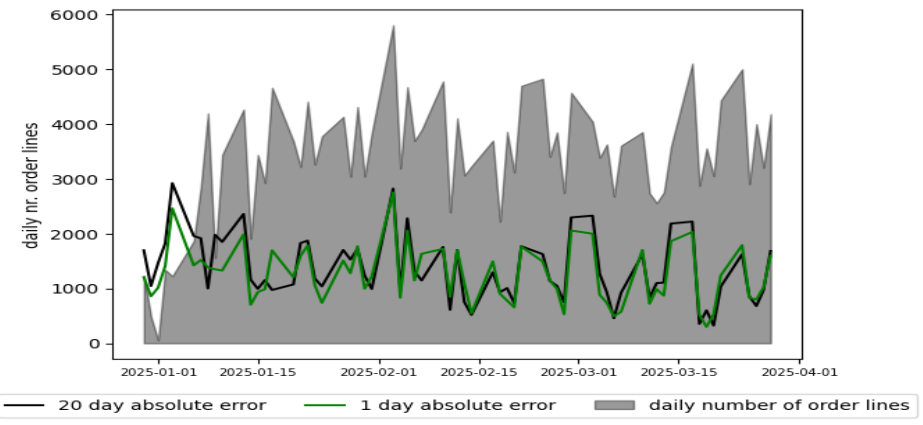


Figure 47: 1 and 20 day ahead absolute errors in wk1 - wk13 2025

Model settings:

model=NeuralProphet(n_forecasts=20,yearly_seasonality=True[4],weekly_seasonality=True,daily_seasonality=False,normalize="soft",optimizer="AdamW",loss_func="L1",batch_size=128,learning_rate=0.1 epochs=201)

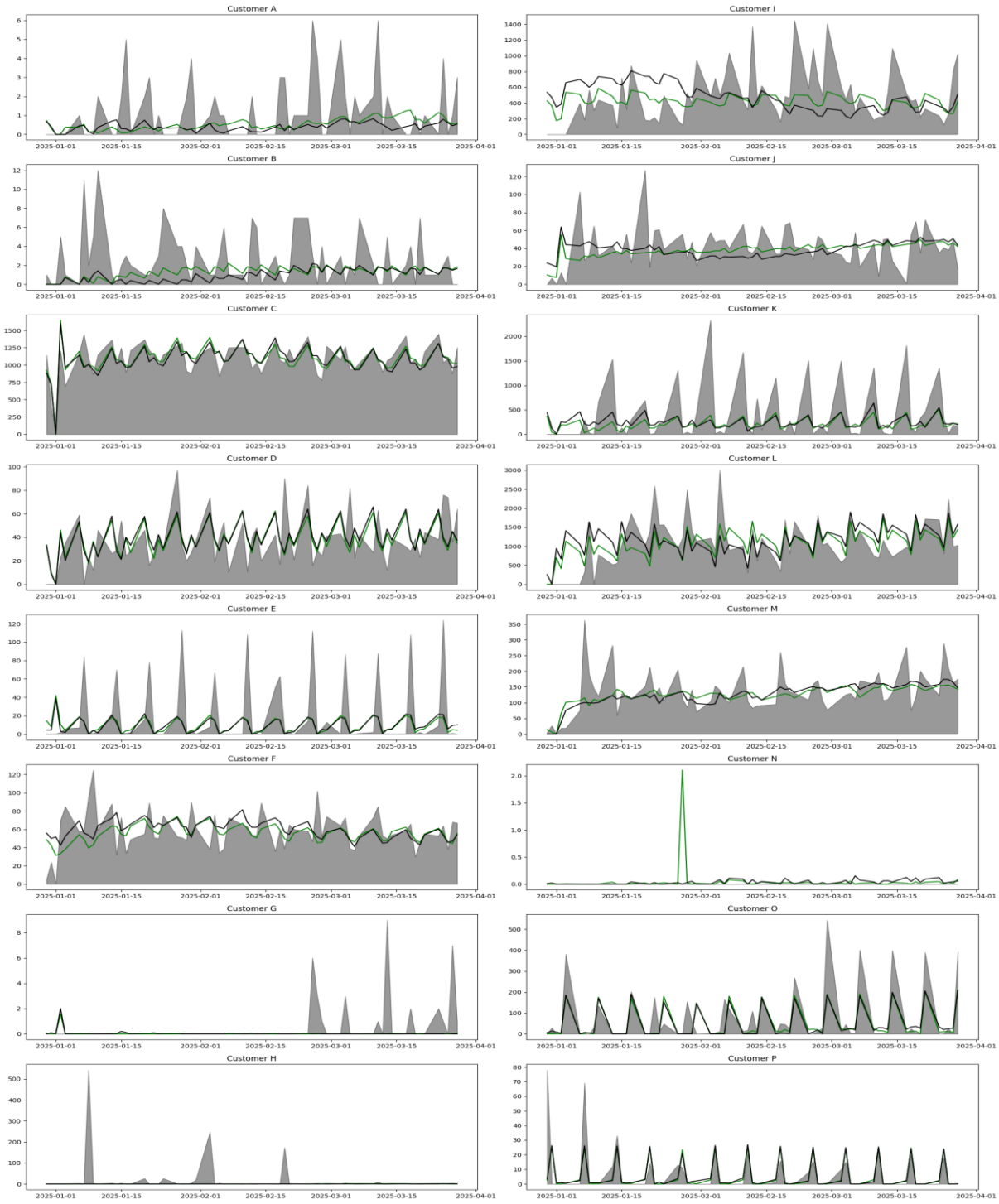


Figure 48: 1 (green) and 20 (black) day ahead predictions for test period Prophet with holidays

[4] As mentioned in the report yearly seasonality is turned of for customer H due to lack of data

A.13 Results (Neural)Prophet with overviews

NeuralProphet with overviews as lagged regressors test period results

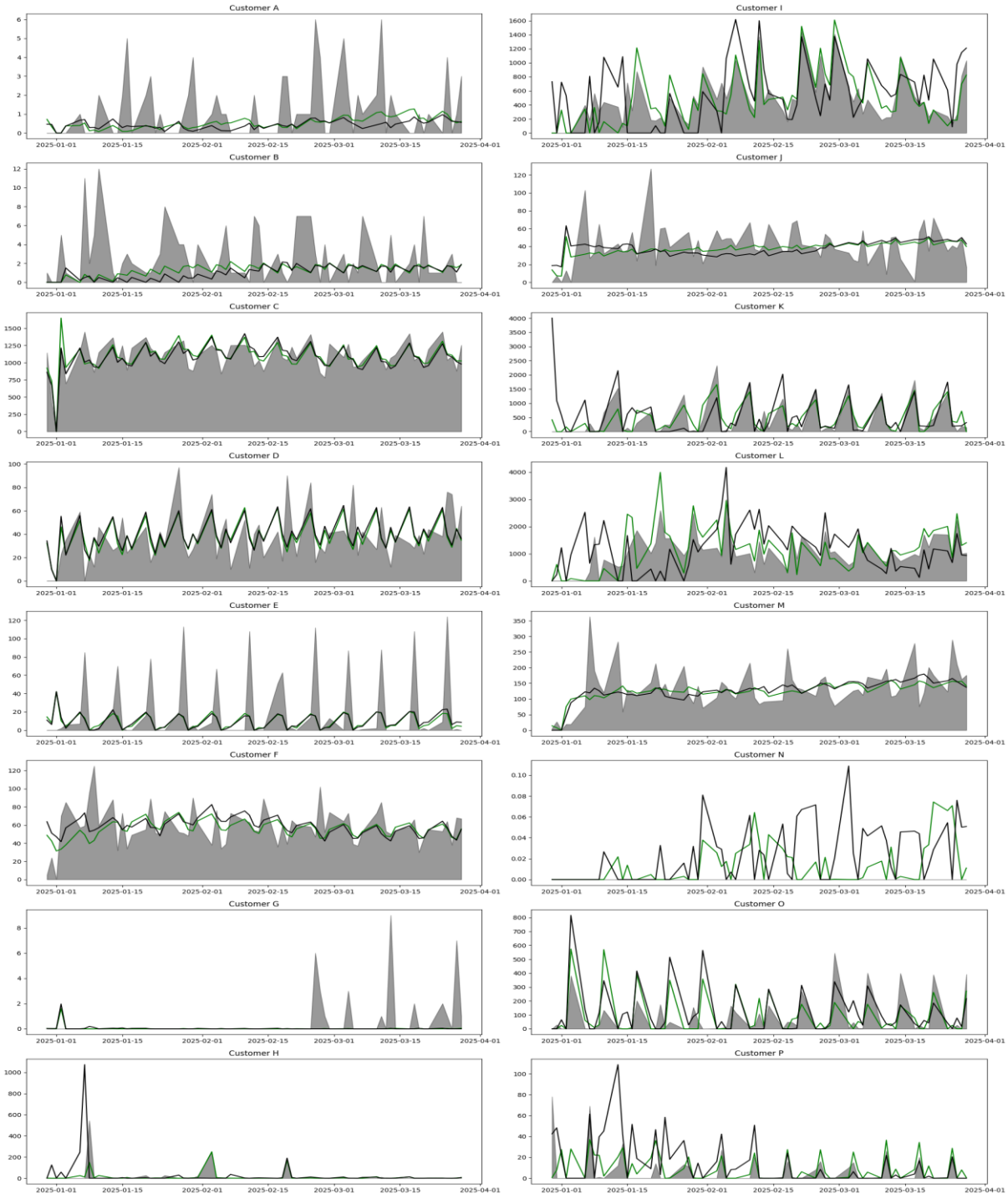


Figure 49: 1 (green) and 20 (black) day ahead predictions for test period NeuralProphet with overviews as lagged regressors

Customer bias (%) Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	56.8	57.1	57.7	58.1	58.8	58.7	59.1	59.7	59.7	61.3	61.6	62.2	63.5	63.4	64.7	65.3	65.7	66.5	66.0	67.6
B	49.4	49.9	50.9	51.8	52.6	54.2	54.4	55.2	55.7	56.7	58.0	57.7	58.7	59.0	59.6	60.7	60.5	61.4	61.3	62.2
C	0.0	0.6	0.2	0.2	0.4	0.4	0.4	0.4	0.4	0.6	0.7	0.8	1.0	1.0	1.5	0.8	0.8	1.0	0.9	1.1
D	-3.1	-3.5	-3.7	-4.3	-4.2	-4.8	-5.1	-5.4	-5.8	-6.1	-7.0	-7.0	-7.1	-8.2	-7.5	-7.4	-7.4	-7.8	-7.7	-7.7
E	53.9	53.6	53.7	53.9	53.4	53.5	53.5	53.6	52.6	50.6	52.9	53.4	51.8	52.4	52.6	53.5	53.9	54.1	54.2	54.5
F	5.8	5.7	5.2	5.1	4.7	4.4	4.2	3.6	3.2	2.4	1.9	1.9	1.2	1.0	0.5	0.1	-0.0	-0.9	-1.1	-1.3
G	94.5	93.4	93.3	93.2	92.9	93.0	94.5	92.0	94.0	92.2	92.1	92.0	91.7	91.7	91.8	91.6	91.5	91.3	91.0	91.2
H	33.6	-10.7	-12.6	-0.9	-11.3	-4.3	-11.7	-6.4	-28.8	-70.4	-89.2	-117.3	-18.0	-27.0	-70.8	-320.4	33.7	61.7	15.1	-19.4
I	0.2	-0.5	-4.0	-7.2	-6.3	-9.0	-6.0	-7.4	-12.5	-14.3	-11.7	-8.4	-17.5	-14.8	-14.7	-23.1	-11.5	-14.1	-16.5	-2.4
J	7.6	6.8	7.3	7.6	2.4	7.0	7.5	7.8	7.8	7.6	7.4	7.2	7.3	7.0	6.8	6.3	5.8	5.6	5.1	4.8
K	-8.1	-1.9	-4.9	-6.9	-6.9	-19.5	-15.3	-8.8	3.4	-29.7	-32.0	-27.5	-33.7	-32.2	-31.9	-9.1	-4.6	-28.4	-34.5	-41.3
L	-9.6	-8.1	-10.4	-10.0	-13.7	-14.9	-11.5	-10.3	-10.6	-10.4	-12.3	-8.5	-15.5	-4.9	-6.3	-4.2	1.4	-1.7	-5.8	-2.8
M	7.3	6.9	7.2	7.2	7.0	7.0	6.6	6.5	6.5	6.2	5.9	5.6	5.6	6.0	5.9	6.1	6.1	6.2	6.6	6.8
N	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
O	-13.3	-26.7	-35.1	-41.4	-24.3	-31.3	-32.2	-54.8	-38.2	-46.6	-56.1	-43.2	-51.2	-39.1	-51.5	-44.5	-49.3	-79.7	-80.4	-101.6
P	-49.3	-71.1	-75.9	-65.2	-98.2	-105.4	-144.3	-118.9	-123.5	-152.5	-130.5	-135.4	-89.5	-93.1	-124.7	-140.9	-161.2	-152.8	-183.4	-185.1

Table 22: Customer bias (%) NeuralProphet with overviews as lagged regressors for all days in advance

Customer MAD Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.1
B	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3
C	136.2	128.6	130.9	133.1	131.9	136.9	131.5	133.4	129.6	136.1	141.6	133.1	131.5	129.1	127.5	135.9	134.1	132.4	133.6	137.1
D	17.4	16.9	17.2	16.8	16.7	17.2	16.8	17.1	16.9	17.1	17.3	17.2	17.1	17.7	17.2	17.3	17.0	17.4	16.9	17.2
E	19.2	19.2	19.1	19.0	19.0	19.2	19.0	19.2	19.1	19.7	19.3	19.1	19.6	19.3	19.4	19.6	19.3	19.4	19.1	19.1
F	16.3	16.7	17.1	17.4	16.8	16.8	16.9	16.9	17.0	16.5	16.6	16.2	16.5	16.2	17.1	16.4	16.8	16.6	16.1	17.5
G	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
H	9.6	6.3	6.2	4.5	6.3	4.5	6.2	5.4	8.9	15.8	18.1	22.7	12.8	11.5	39.3	68.8	21.2	21.9	28.0	34.9
I	146.3	147.1	175.9	200.5	203.0	218.3	228.3	256.5	271.8	293.3	267.4	275.8	290.1	289.4	326.6	349.5	373.9	381.3	400.1	389.0
J	15.5	16.1	15.9	15.7	18.4	16.6	16.5	16.4	16.4	16.9	16.8	16.9	17.3	16.9	17.5	17.6	17.8	18.0	17.8	18.1
K	218.9	193.1	241.0	222.5	229.4	264.7	274.7	231.5	283.9	245.9	276.7	276.9	250.8	276.9	321.3	350.5	340.4	448.9	417.2	427.1
L	344.4	318.1	368.1	385.2	446.5	483.1	518.5	559.9	608.6	618.1	642.8	673.5	683.9	728.6	725.0	678.6	755.4	818.7	806.1	836.0
M	43.4	43.9	43.5	44.0	43.0	41.6	42.2	41.6	43.0	43.0	43.6	43.7	42.2	42.2	42.4	40.7	39.9	41.6	40.6	42.3
N	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
O	69.0	77.3	82.8	81.4	74.9	78.9	81.5	92.4	80.5	95.7	101.6	87.8	88.8	79.9	86.6	90.9	85.4	94.2	94.9	125.1
P	7.6	7.1	7.9	8.1	9.3	10.8	10.7	10.4	10.7	12.8	11.2	11.8	8.8	10.7	10.7	12.8	13.3	15.0	13.9	15.8

Table 23: Customer MAD NeuralProphet with overviews as lagged regressors for all days in advance

Prophet with overviews as future regressors test period results

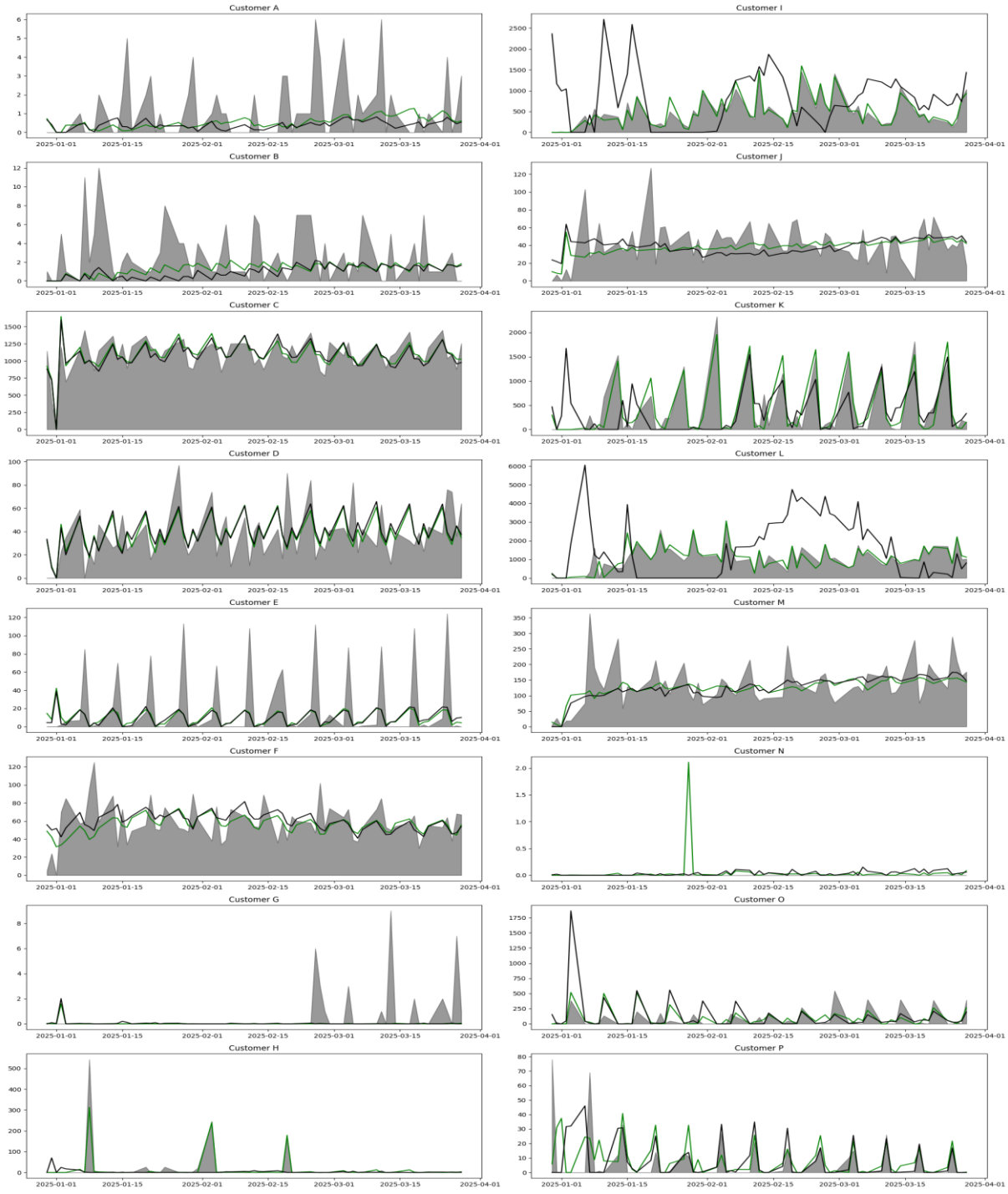


Figure 50: 1 (green) and 20 (black) day ahead predictions for test period NeuralProphet with overviews as future regressors

Customer bias (%) Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	56.8	57.1	57.7	58.1	58.8	58.7	59.1	59.7	59.7	61.3	61.6	62.2	63.5	63.4	64.7	65.3	65.7	66.5	66.0	67.6
B	49.4	49.9	50.9	51.8	52.6	54.2	54.4	55.2	55.7	56.7	58.0	57.7	58.7	59.0	59.6	60.7	60.5	61.4	61.3	62.2
C	0.0	0.6	0.2	0.2	0.4	0.4	0.4	0.4	0.4	0.6	0.7	0.8	1.0	1.0	1.5	0.8	0.8	1.0	0.9	1.1
D	-3.1	-3.5	-3.7	-4.3	-4.2	-4.8	-5.1	-5.4	-5.8	-6.1	-7.0	-7.0	-7.1	-8.2	-7.5	-7.4	-7.4	-7.8	-7.7	-7.7
E	53.9	53.6	53.7	53.9	53.4	53.5	53.5	53.6	52.6	50.6	52.9	53.4	51.8	52.4	52.6	53.5	53.9	54.1	54.2	54.5
F	5.8	5.7	5.2	5.1	4.7	4.4	4.2	3.6	3.2	2.4	1.9	1.9	1.2	1.0	0.5	0.1	-0.0	-0.9	-1.1	-1.3
G	94.5	93.4	93.3	93.2	92.9	93.0	94.5	92.0	94.0	92.2	92.1	92.0	91.7	91.7	91.8	91.6	91.5	91.3	91.0	91.2
H	24.1	21.2	18.1	19.7	11.4	2.1	8.2	-3.4	-9.8	-24.6	-61.9	2.6	57.7	60.7	68.5	71.5	90.2	48.6	78.9	82.0
I	0.8	0.9	-4.1	-6.4	-6.4	-8.2	-6.3	-5.3	-8.1	-3.2	-16.3	-7.3	0.7	-20.1	-9.5	-24.7	-34.7	-61.9	-64.8	-68.3
J	7.3	6.9	7.1	7.0	7.5	7.3	7.2	7.6	7.5	7.5	7.5	7.2	7.3	7.0	6.8	6.4	5.8	5.8	5.4	5.0
K	0.7	4.4	-1.9	-9.0	-11.7	-11.6	-17.6	-18.3	-26.5	-38.2	-25.7	-25.7	-39.7	-57.9	-69.1	3.4	-11.7	-48.1	10.0	11.1
L	-0.3	2.1	1.2	3.2	0.6	0.1	-2.8	-9.7	-10.9	-2.7	-1.1	-4.4	-6.5	-7.2	-2.4	-4.7	-30.6	-10.0	-13.4	-40.2
M	7.5	7.3	7.4	7.3	7.4	7.1	7.0	6.9	6.8	6.8	6.6	6.5	6.4	6.5	6.6	6.7	6.6	6.8	7.1	7.6
N	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf	-inf
O	-11.3	-19.2	-19.0	-41.1	-15.6	-25.1	-29.2	-45.0	-42.4	-54.9	-51.7	-49.0	-59.5	-16.6	-51.4	-171.3	-292.8	-435.5	-449.7	-42.9
P	-34.0	1.7	-23.4	-5.4	-27.2	-29.3	-30.9	-22.4	-37.1	-18.5	-46.5	-66.6	-34.0	-53.4	-18.5	-105.9	20.0	-123.8	-14.0	-15.7

Table 24: Customer bias (%) Prophet with overviews as future regressors for all days in advance

Customer MAD Table

ID	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.2	1.1
B	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.2	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3	2.3
C	136.2	128.6	130.9	133.1	131.9	136.9	131.5	133.4	129.6	136.1	141.6	133.1	131.5	129.1	127.5	135.9	134.1	132.4	133.6	137.1
D	17.4	16.9	17.2	16.8	16.7	17.2	16.8	17.1	16.9	17.1	17.3	17.2	17.1	17.7	17.2	17.3	17.0	17.4	16.9	17.2
E	19.2	19.2	19.1	19.0	19.0	19.2	19.0	19.2	19.1	19.7	19.3	19.1	19.6	19.3	19.4	19.6	19.3	19.4	19.1	19.1
F	16.3	16.7	17.1	17.4	16.8	16.8	16.9	16.9	17.0	16.5	16.6	16.2	16.5	16.2	17.1	16.4	16.8	16.6	16.1	17.5
G	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6	0.6
H	6.6	6.6	5.7	5.9	5.8	4.8	4.4	3.9	4.9	7.5	13.9	4.2	12.1	12.5	13.7	15.6	18.0	24.9	18.1	19.4
I	65.5	69.6	89.6	102.7	117.0	155.9	161.2	207.9	202.8	209.5	258.4	213.7	231.2	263.0	276.5	379.2	417.0	573.1	590.8	603.4
J	15.8	15.9	15.9	16.0	16.1	16.3	16.2	16.3	16.2	16.4	16.9	16.7	17.0	16.9	17.3	17.7	17.2	17.5	17.5	18.1
K	145.8	142.3	160.8	198.0	215.5	226.8	236.2	253.2	308.9	339.2	289.4	282.7	302.6	374.0	397.3	422.6	432.6	655.6	492.0	326.1
L	181.3	206.9	288.8	338.1	371.0	384.2	479.7	559.4	571.1	572.3	565.4	651.5	666.4	670.0	721.3	771.4	1079.1	1019.6	1193.9	1422.4
M	43.2	43.2	44.3	44.2	42.1	42.5	42.0	42.7	42.8	43.3	44.2	43.4	42.5	43.7	43.4	41.3	41.2	41.5	40.4	40.4
N	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0
O	68.5	75.6	71.3	83.5	59.9	77.4	79.2	87.2	86.2	101.0	99.6	97.5	104.9	76.3	87.0	173.9	264.3	355.4	354.9	96.6
P	7.1	6.0	6.0	6.1	6.1	6.8	6.0	6.7	7.5	7.1	6.5	6.4	6.2	8.0	6.4	11.2	10.6	17.8	10.5	6.1

Table 25: Customer MAD Prophet with overviews as future regressors for all days in advance

Customer level results

Table 29 shows the MAD percentage improvement of the three different order overview inclusion methods discussed in Section 4.4 compared to the reconstructed method, as well as the model bias, as percentage of the total demand.

		days ahead	***H***	***I***	***K***	***L***	***O***	***P***
MAD (% improvement)	**Overviews lagged**	7 days	-37.8	-184.7	-34.3	-136.3	3.5	-63.0
		20 days	-155.1	-101.9	-30.7	-108.8	-8.3	-103.7
	Overviews future	7 days	-26.7	-64.4	-8.2	-85.7	8.9	-16.7
		20 days	-50.7	-91.0	-39.9	-121.2	-54.7	-44.4
	Overviews direct	7 days	0.0	0.0	0.0	0.0	-	-
		20 days	8.7	-1.0	-4.4	1.3	-	-
Bias (%)	**Overviews lagged**	7 days	-2.6	-4.7	-9.1	-11.2	-29.2	-87.1
		20 days	-33.8	-10.1	-18.7	-8.5	-47	-120
	Overviews future	7 days	15	-4.2	-6.7	0.6	-22.9	-21.2
		20 days	28.3	-17.7	-19.2	-7	-96.2	-34.2
	Overviews direct	7 days	8.2	0.2	5.4	-8.1	-	-
		20 days	21.8	-5.9	19.5	-11.9	-	-

Table 26: Customer performance three overview inclusion methods

A.14 ACF plot per customer

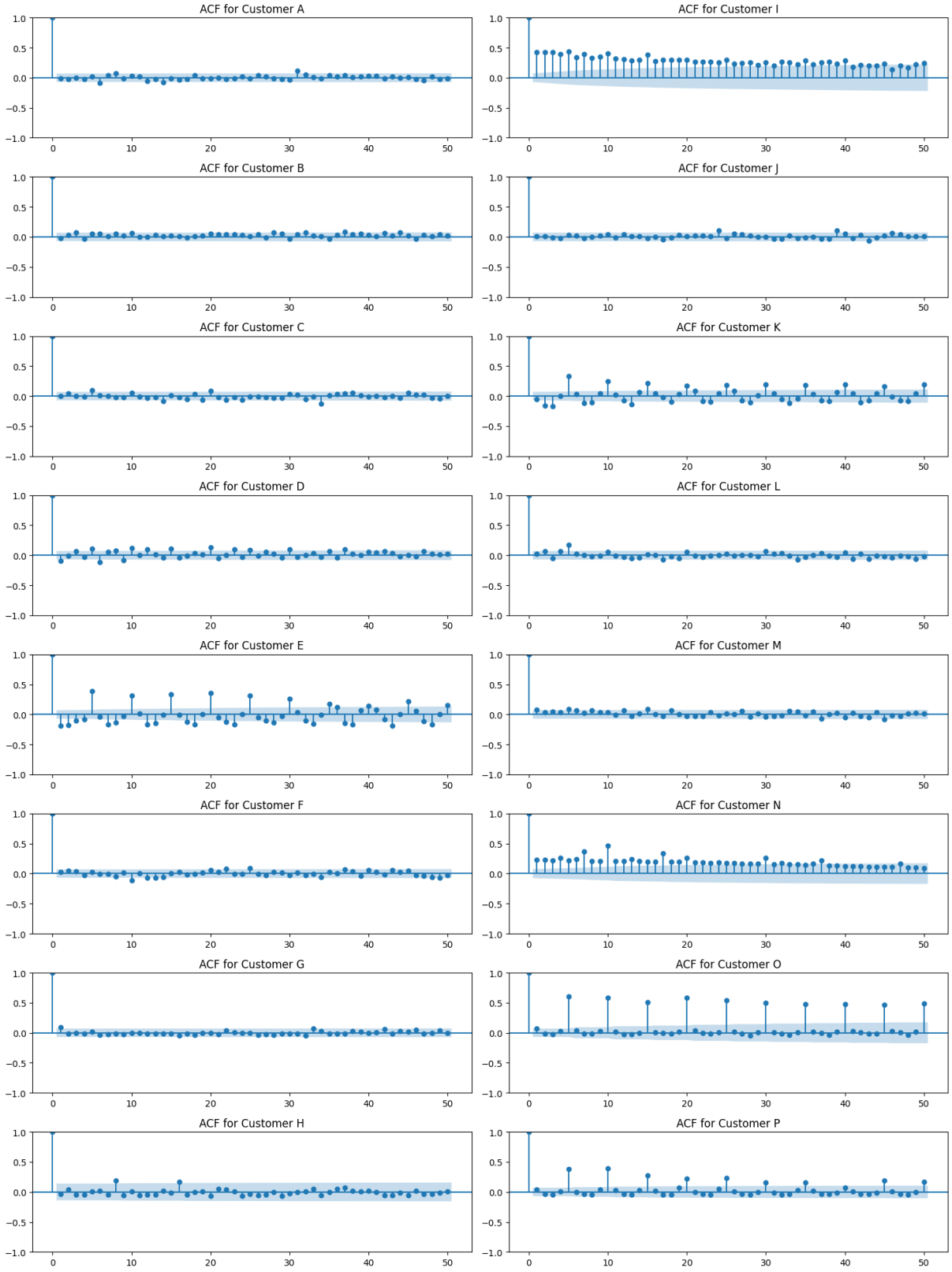


Figure 51: ACF per customer

A.15 Results NeuralProphet with (deep) AR

	MAD		MAD (% improvement compared to overviews direct)				Bias (%)		Bias (% improvement compared to overviews direct)			
	Overviews direct		AR		Deep AR		Overviews direct		AR		Deep AR	
days ahead	7 days MAD	20 days MAD	7 days MAD	20 days MAD	7 days MAD	20 days MAD	7 days bias	20 days bias	7 days bias	20 days bias	7 days bias	20 days bias
A	1.2	1.2	0.0	0.0	0.0	0.0	58	61.7	18.3	19.0	5.5	-1.4
B	2.2	2.3	-4.5	-4.3	-9.1	-4.3	51.9	56.5	13.3	9.6	29.3	17.7
C	132.7	133.2	1.1	-0.6	-2.4	-4.4	0.3	0.7	-33.3	-71.4	-66.7	-233.3
D	17	17.1	8.8	5.3	1.2	-3.5	-4.1	-6	-14.6	-3.3	-48.8	-112.2
E	19.1	19.2	33.5	29.7	45.5	43.2	53.6	53.3	84.1	84.1	84.7	86.9
F	16.9	16.7	6.5	1.8	4.7	3.0	5	2.4	34.0	83.3	14.0	46.0
G	0.6	0.6	0.0	0.0	0.0	0.0	93.5	92.4	-0.1	1.2	-0.3	-0.4
H	4.5	6.3	0.0	0.0	0.0	-1.6	8.2	21.8	0.0	3.2	0.0	-148.8
I	66.2	137.2	0.0	-2.3	0.0	-8.5	0.2	-5.9	0.0	-6.8	0.0	-4000.0
J	16	16.7	-14.4	-12.6	-8.7	-9.0	7.2	6.9	-109.7	-97.1	-100.0	-77.8
K	174.9	231.3	0.0	1.0	0.0	1.6	5.4	19.5	0.0	4.6	0.0	-192.6
L	173.1	283.8	0.0	1.1	0.0	1.3	-8.1	-11.9	0.0	0.8	0.0	-58.0
M	43.1	42.6	-8.6	-11.7	-4.4	-7.0	7.3	6.9	74.0	82.6	42.5	49.3
N	0	0	0.0	0.0	0.0	0.0	-	-	-	-	-	-
O	46.1	47.1	-16.7	-38.2	-21.0	-39.1	38.4	37.5	59.4	35.2	99.2	79.7
P	4.3	4.4	0.0	2.3	-4.7	-4.5	12.5	10.8	-18.4	-46.3	-3.2	8.0

Table 27: Test performance (deep) AR compared to the T,S,E with order overview replacement model on customer level

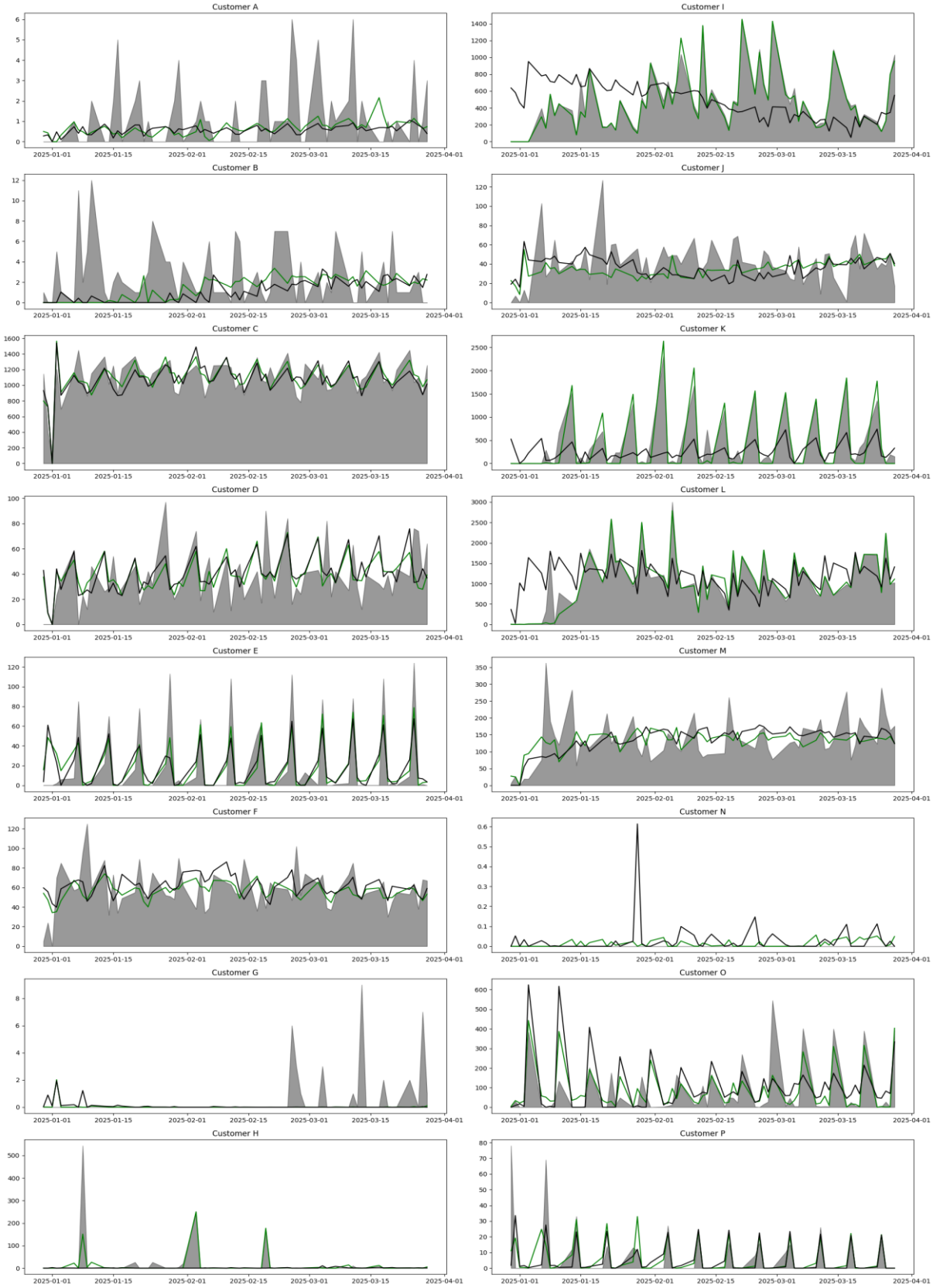


Figure 52: Predictions test set with AR (n_lags=20)

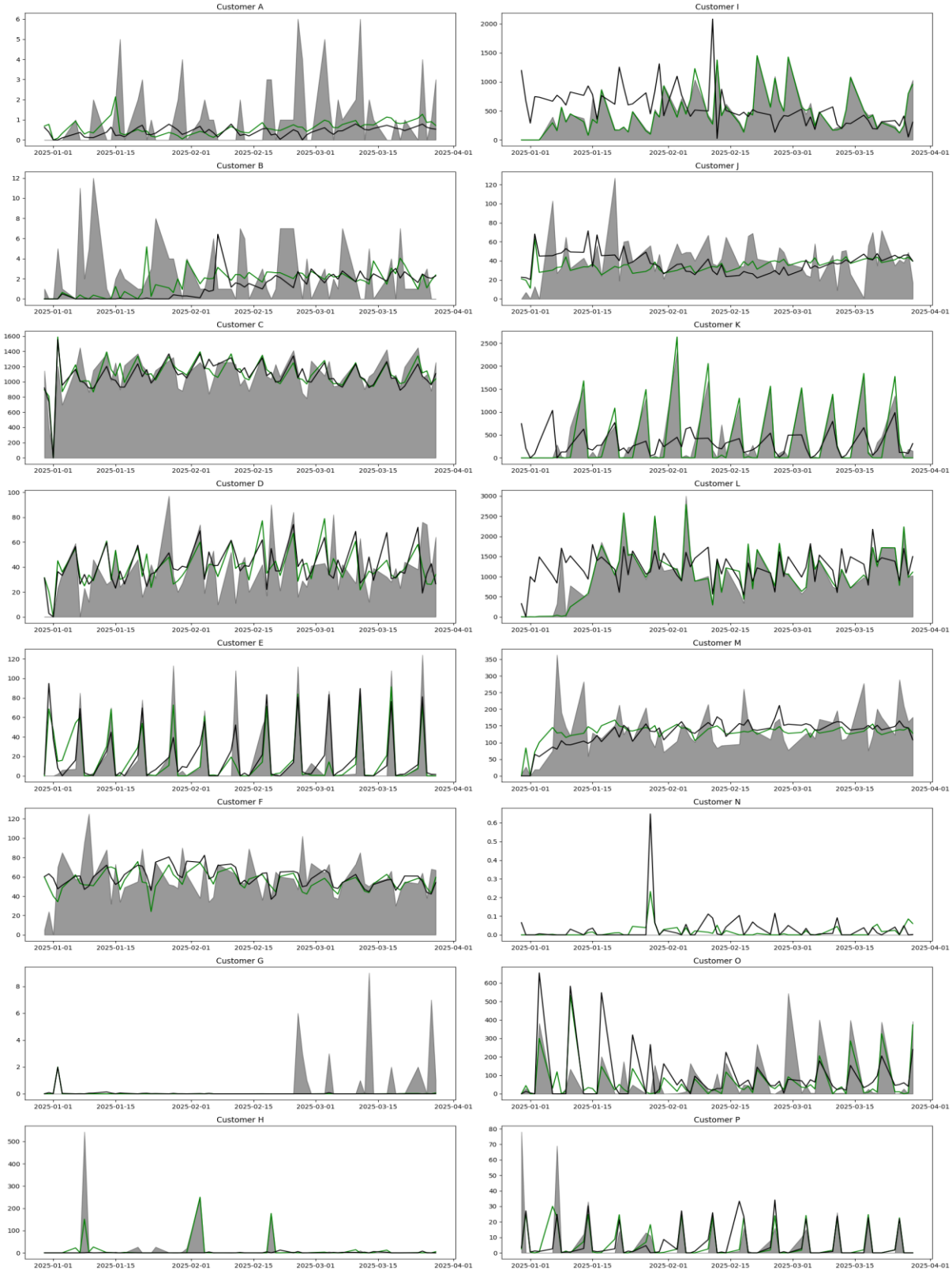


Figure 53: Predictions test set with Deep AR (n_lags=20 & 1 hidden layer with 20 nodes)

A.16 Results final model

	MAD				MAD (% improvement)		Bias (%)				Bias (% improvement)	
	Reconstruction		Final model		Final model		Reconstruction		Final model		Final model	
days ahead	7 days MAD	20 days MAD	7 days MAD	20 days MAD	7 days MAD	20 days MAD	7 days bias	20 days bias	7 days bias	20 days bias	7 days bias	20 days bias
A	1.8	1.8	1.2	1.2	33.3	33.3	-7.6	-7.6	47.4	50	-523.7	-557.9
B	2.7	2.7	2.3	2.4	14.8	11.1	30.7	30.7	40.2	44.4	-30.9	-44.6
C	257.4	257.4	131.8	133.7	48.8	48.1	-8.5	-8.5	0.5	1.3	94.1	84.7
D	25.4	25.4	15.5	16.2	39.0	36.2	-11.9	-11.9	-4.6	-6.5	61.3	45.4
E	32	32	9.6	10.4	70.0	67.5	-21.3	-21.3	6.5	7.1	69.5	66.7
F	45.7	45.7	16.1	16.4	64.8	64.1	-57.9	-57.9	3.3	0.6	94.3	99.0
G	1.5	1.5	0.6	0.6	60.0	60.0	-86.1	-86.1	93	92.2	-8.0	-7.1
H	4.5	6.9	4.5	6.3	0.0	8.7	8.2	18.2	8.2	21.8	0.0	-19.8
I	66.2	135.8	66.2	136.2	0.0	-0.3	0.2	0.6	0.2	-5.6	0.0	-833.3
J	24	24	16.1	16.7	32.9	30.4	31.7	31.7	7.7	7.3	75.7	77.0
K	174.9	221.6	174.9	228.4	0.0	-3.1	5.4	11.5	5.4	18.4	0.0	-60.0
L	173.1	287.4	173.1	280.9	0.0	2.3	-8.1	-9.7	-8.1	-11.9	0.0	-22.7
M	80.5	80.5	46.3	46.6	42.5	42.1	-47.7	-47.7	3.5	3.2	92.7	93.3
N	0	0	0	0	0.0	0.0	-	-	-	-	-	-
O	80.8	80.8	46.1	47.5	42.9	41.2	-50.4	-50.4	37.4	36.8	25.8	27.0
P	5.4	5.4	4.3	4.4	20.4	18.5	16.2	16.2	12	10.4	25.9	35.8

Table 28: Customer performance comparison between final model and reconstruction

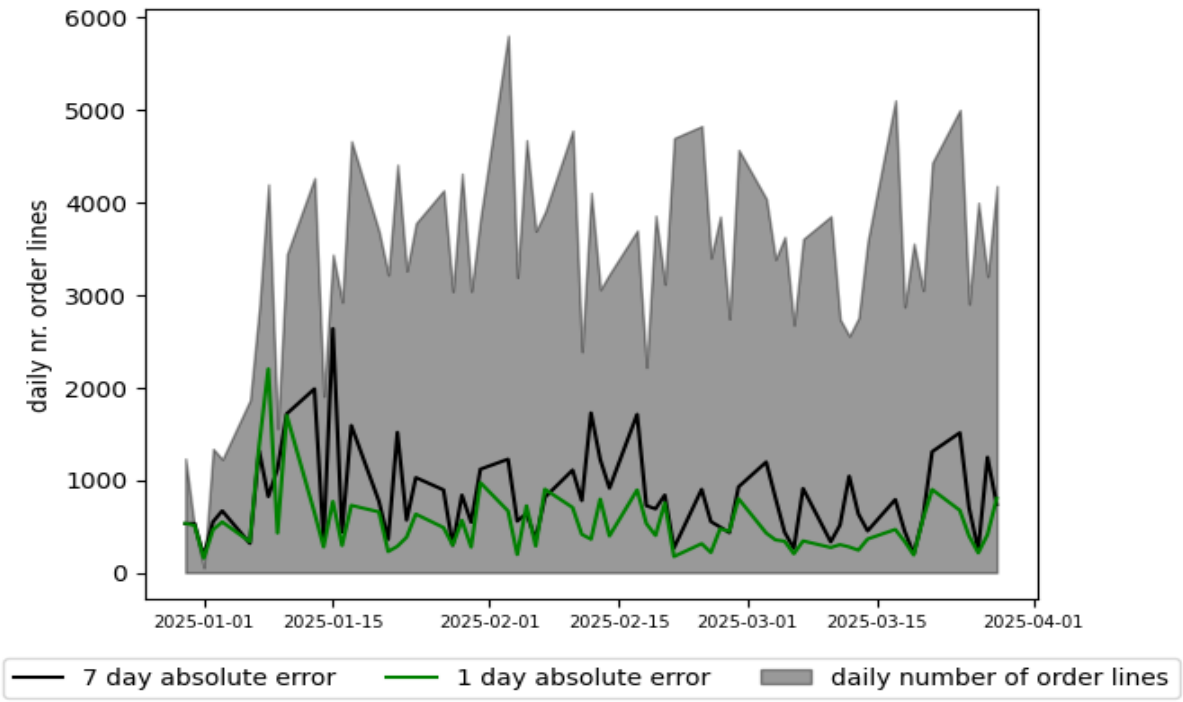


Figure 54: 1- and 7-day prediction errors of final model on test set

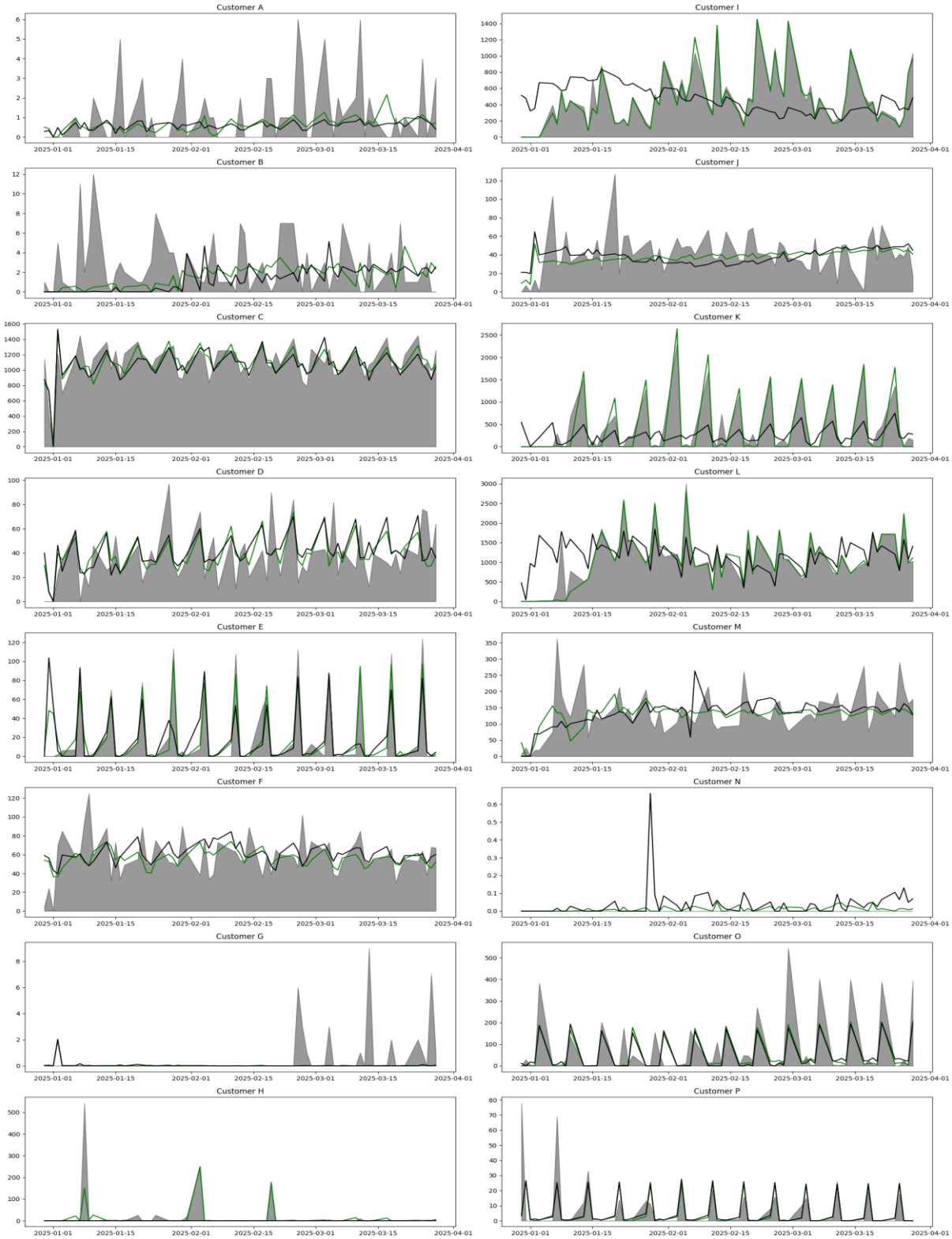


Figure 55: Predictions test set with final model

MAPE (%) per step	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*	*17*	*18*	*19*	*20*
Reconstructed model	60.9	61.0	64.8	64.4	64.2	66.0	65.4	65.9	65.7	67.4	67.6	67.5	68.3	69.2	68.3	71.5	71.2	139.2	141.9	145.1
Final model	10.9	11.2	13.3	14.4	14.3	15.9	16.2	16.1	16.0	17.0	17.7	17.0	18.8	19.7	18.4	22.9	23.3	59.3	64.6	64.7

Table 29: MAPE of reconstructed and final model on aggregated level per time step test set

A.17 Change management meeting concepts

Suggested participants for the following sessions:
- Team management: responsible for supporting the vision and ensuring its implementation is controlled.
- Team leaders: responsible for relocating workload, daily schedules, relocating staff, discussing overtime, early leave, etc.

Vision information session

In the vision information session, the affected employees are informed about the vision and its contents, which includes the proposed method. After the session, the employees should know the vision, background, benefits, and planned next steps for implementation. It should work as follows:

1. Change initiators provide the following information:
   1.1 Background leading to the development.
   1.2 Contents of the vision.
   1.3 Planned next steps
2. Opportunity to answer questions.

Note: Stolzenberg & Heberle (2022) recommend the presentation is given by the change initiators as well as some managers, to make clear that the whole management team supports the vision.

1.1	**Procedure** Discuss the background of the vision. This includes highlighting the current procedure that includes frequent relocation of staff, overworking and lack of overall control on the workload.
1.2	**Procedure** Introduce the vision: "A good operations environment can only be achieved when there is a level of control and calmness around the scheduled and required capacity." Introduce the implication: It implies that there is the need to put greater emphasis and focus on aligning the scheduled with the required capacity. One aspect required for that is obtaining greater insight in future demand, which can be done through the demand forecast. Other aspects involve obtaining greater insight in available capacity and possibilities to redistribute workload across the days according to internal processing rules. To successfully realise the vision these aspects, need to be combined and integrated. Introduce the new method, its application, and results. And highlight how this new method can help realise the vision discussed before by providing greater insight in the daily workload.
1.3	**Procedure** Discuss the next steps. These include the vision dialog kick-off and continuous vision dialogs.
2	**Procedure** In pairs, participants write questions they would like to ask the initiators regarding the vision background, content, and next steps. Panel discussion

Vision dialog kick-off

During the kick-off, the contents of the vision, and its implications for the work of the participants will be discussed. In particular it should serve to show the practical value of the vision (Stolzenberg & Heberle, 2022). We suggest doing so by using case scenarios that illustrate the practical value through interactive participation. We suggest the following content for this session:

1. Recap content of the vision information session.
2. Discuss the vision, and its relation to their work.
3. Play case scenario.
   3.1 Illustrate current bottlenecks by providing case and playing how its currently solved.
   3.2 Introduce method output and discuss together how it may help tackling the bottlenecks observed before.
   3.3 Solve the case using introduced method.
4. Discuss how the new solution can be further developed and the next steps.

1	**Procedure** Recap previous information session.
2	**Procedure** In pairs, participants write down how the vision relates to their work, and what they can do in their role to contribute to the realisation of the vision. These may include, but are likely not limited to: - Creating staff and workload schedules that are aligned with each other. - Relocating orders to other days to ensure long-term alignment of daily staff and workload. - Relocating staff to other stations to ensure alignment of daily staff and workload (but relocating on short notice may result in negative feedback from staff). - Exploring possibilities for overtime work, ahead of the realisation in case this is needed. Discuss the mentioned contributions and list them on a whiteboard.
3.1	**Procedure** Introduce the following case: We are going to play out our process for the normal picking division for week X and try to realise this vision. We may use the actions that we just listed and play according to the rules. The rules are a (slightly) simplified version of reality. We use actual input and demand. For the processing rate per order line, we use 25 per staff hour, which is the average observed in 2024. For simplicity, the sick rate is ignored so scheduled hours are realised (in reality the short leave sick rate is approximately 2%). 1. We will make the staff schedule for week X for normal picking. What do we need to make the staff schedule? Then make the schedule together as if it is Thursday. Input that will likely be mentioned: - staff availability week X - VMI order lines to pick from Wednesday - anything else required as input? 2. For Monday to Friday, play the case per day: - Analyse nr. of order lines that need to be finished and staff that we scheduled -> do we finish early or late? - Do we take other actions? (do we do work for next days? or continue working a little longer e.g.)

	Recap: what happened? How were our results, did we face overtimes, what did we already do for next week? Where did it go wrong? Did we forget inventory counts or did we do those as well? Possible things that could have gone wrong: - Observing large fluctuations in daily workload due to not redistributing the workload effectively across the workdays. - Not noticing days that will cause problems in time and not looking ahead causing overtime on certain days without being able to act for these days in advance.
3.2	**Procedure** Show the results of the demand forecast on the day of scheduling, and what insight it can give in the workload and how it can be rescheduled using the demand forecast output as a Yamazumi chart that highlights work that can be rescheduled to other days (using the information from the internal processing rules overview).
3.3	**Procedure** Play the case again, now using the demand forecast as well. 3. We will make the staff schedule for week X for normal picking. What do we need to make the staff schedule? Then make the schedule together as if it is Thursday. - staff availability week X - Demand forecast - anything else required as input? *Make a concept for how we can distribute the workload across the days in a way that balances the daily workload, and make the staff schedule.* 4. For Monday to Friday, play the case per day: - Analyse nr. of order lines and staff that we scheduled -> do we finish early or late? - Analyse Demand forecast - Do we take other actions? (such as revising the workload / staff schedule) (do we do work for next days? or continue working a little longer e.g.) Recap: what happened? How were our results compared to previous time, did we face overtime, what did we already do for next week? Where did it go wrong? Did we forget inventory counts or did we do those as well?
4	**Procedure** Discuss whether using the demand forecast and creating a workload schedule next to a staff schedule improved the results compared to the results before. Discuss in pairs how this can be done in practice and what else is required next to the demand forecast to do this effectively. Ideas may be, but are not limited to: - Making an automatic workload schedule based on the demand forecast and internal processing rules. - Converting the workload schedule to required staff hours. - Create a method that estimates the net hours based on the gross hours that are scheduled (so excluding short term sick leave, and estimated lunch breaks). - Extending the forecast to inbound and outbound divisions. Discuss findings altogether. Discuss how to proceed:

	1. Continuous vision dialogs to exchange vision implementation, progress, challenges, further required developments, and to keep the vision alive. 2. Other actions to be performed such as exploring new methods and/or actions that were identified in previous step.

Continuous vision dialogs

Continuous vision dialogs may serve to keep track of the implementation progress, make changes when required, and keep the vision alive. They could be held every few weeks at first and eventually be conducted every few months (Stolzenberg & Heberle, 2022). We suggest the following content for this session based on the examples provided by Stolzenberg & Heberle:

1. Recap content of the previous session.
2. Discuss progress and challenges of the implementation.
3. Determine actions that should be started, kept, and stopped to improve the implementation.

1	**Procedure** Recap findings previous session.
2	**Procedure** Discuss progress and challenges of the implementation together.
3	**Procedure** Let the participants answer the following questions: 1. What should be extended / started to create more control and calmness around the scheduled and required capacity? (yellow cards) 2. In what ways do we already create control? (green cards) 3. What is standing in the way, so should be stopped or changed? (red cards) Place the cards on a whiteboard and discuss them together to make a list of actionable points for next time.