Machine Learning as a Tool for Demand Forecasting:

A Study at HMS Networks

Authors:
Jacob Jhaveri & Måns Skogman

Supervisor at HMS Networks: Alexander Fröström
Supervisor at LTH: Eva Berg
Examiner at LTH: Andreas Norrman

June 2026

Master Thesis in Industrial Engineering and Management

Division of Engineering Logistics, Department of Industrial and Mechanical Sciences
Faculty of Engineering, LTH, Lund University, PO Box 118, SE-221 00 LUND, Sweden
Lund 2026

Acknowledgements

This thesis was conducted during the Spring of 2026 by two authors as a final project of a master at the Faculty of Engineering at Lund University. The project was conducted at the Division of Engineering Logistics, Department of Industrial and Mechanical Sciences. The thesis dealt with the development of machine learning models as a demand forecasting tool using a design science research methodology. The models were applied and evaluated using data from the case company, HMS Networks.

We would like to thank our supervisor at HMS Networks, Alexander Fröström, for providing continuous feedback and support throughout the project. Furthermore, we want to express our thanks to Joakim Nideborn for initiating the project at HMS Networks and assisting during the start-up of the thesis.

Additionally, we want to thank our supervisor at the Division of Engineering Logistics, Eva Berg, who has offered helpful guidance throughout the execution of the project.

Lastly, we would like to thank the interviewees for sharing helpful insights, which made the project possible.

Abstract

This thesis investigates the applicability of using machine learning for demand forecasting at HMS Networks. Using the design science research methodology, multiple forecasting artifacts were developed and tested iteratively, using historical sales data. Four tree-based models; Decision Tree, Random Forest, LightGBM and XGBoost, were tested with different pre-processing and feature engineering techniques such as first-order differencing, scaling and varying levels of granularity. The results show that performance was primarily improved through better data representations, rather than through model selection alone. Particularly, first-order difference and scaling improved the models' abilities to generalize and reduced forecasting bias. Furthermore, the temporal granularity was shown to strongly affect the stability of the forecast, where monthly granularity generated more stable forecasts than weekly. When compared to the current forecast at HMS Networks, XGBoost and LightGBM performed better based on both accuracy and bias. Overall, the findings from this study show that machine learning can be applicable to demand forecasting when the data representation is aligned with the capabilities and limitations of the models.

Author contribution: The authors contributed equally to this thesis.

Keywords: *Demand Forecasting*; *Machine Learning*; *Supply Chain Management*; *Forecasting Metrics*; *Design Science Research*

Acronyms

6M MA	6-Month Moving Average
AI	Artificial Intelligence
APE	Absolute Percentage Error
AR	Autoregression
ARIMA	Autoregression Integrated Moving Average
DDPFF	Dynamic Dual-Phase Forecasting Framework
DSR	Design Science Research
FOD	First Order Differencing
HMLV	High Mix, Low Volume
LightGBM	Light Gradient Boosting Machine
LSTM	Long Short-Term Memory
MA	Moving Average
MAE	Mean Average Error
MAPE	Mean Absolute Percentage Error
ML	Machine Learning
RMSE	Root Mean Squared Error
SARIMA	Seasonal Autoregression Integrated Moving Average
SCM	Supply Chain Management
SKU	Stock Keeping Unit
SIOP	Sales, Inventory and Operations Planning
XGBoost	Extreme Gradient Boosting

Contents

List of Figures

List of Tables

1 Introduction

1.1 Background

The concept of forecasting has existed ever since humans have been interested in predicting the future to facilitate decision-making and planning. The process of forecasting has advanced with quantitative techniques during the 1900s, and continues to advance through the rapid progress in computing which enables analysis of larger and more complex data sets. This development in computer science includes machine learning (ML), which has gained increasing attention from forecasters (Petropoulos et al., 2022). The applications of ML within supply chain management have increased substantially since 2019, regarding both trend and diversity (Babai et al., 2023). Furthermore, the systematic review, conducted by Babai et al. (2023), shows that the field which has attracted the most attention is demand forecasting.

More traditional forecasting methods, such as ARIMA (Autoregression Integrated Moving Average) and exponential smoothing, are however not obsolete, since they are robust and less prone to overfitting compared to more complex methods. Complex models, such as ML-technique XGBoost (eXtreme Gradient Boosting), perform better for more complex patterns (Hammam et al., 2025). For example, for new products, ML has been able to outperform conventional ARIMA (Wang, 2025). Therefore, it is of interest for companies to explore the potential of applying ML-models as demand forecasting solutions. The thesis explores this topic in the context of the company HMS Networks.

1.2 HMS Networks

HMS Networks was founded in Halmstad in 1988 and soon after entered the field of industrial communication. The company helps makers and users of industrial equipment to improve productivity and sustainability by enabling their hardware to communicate with software and systems. The Anybus, which allows automation devices to become network neutral, was released by HMS Networks in 1995 and set the foundation for the escalating growth of the company that followed. During the early 2000s the company expanded internationally as innovators within the field of industrial information and communication technology, leading to HMS Networks currently having operations in over 20 countries and partners in additional 50. The growth continued during the 2010s and forward by acquiring other companies across the globe, with their largest acquisition being the American company Red Lions Controls in 2024. These acquisitions have broadened the product range of HMS Networks and thus increased the complexity of their product portfolio, creating both new opportunities and challenges. One of the challenges is the ability to forecast demand, which is the topic of focus for this thesis.

With fluctuating demand and continuous updates to the product portfolio, HMS Networks faces challenges in producing reliable forecasts to guide production

planning and inventory management. Historically, the consistent growth of HMS Networks has allowed for inaccurate forecasts with positive biases. However, in their current situation, the company has gained an interest in improving their forecasting accuracy. Currently, HMS manages more than 4,500 stock-keeping units (SKUs) and expects the product portfolio to continue expanding. They have three main categories of products; new products, customer-unique products and standard products. Standard products are the products no longer classified as new or customer unique, which is defined as products sold to two or more customers with at least one order every two weeks. The master thesis will focus on the standard products, as this is the largest category, and most relevant to forecast. The filtration of SKUs relevant to the thesis results in a product portfolio of approximately 2000 SKUs.

Currently, demand forecasting is managed through sales representatives in combination with the forecasting tool Anaplan. Sales representatives within each division estimate expected monthly sales over a 12-month horizon for each product category, which are then entered into Anaplan. Anaplan then uses historical data to determine the share of each product category that each individual SKU should constitute, based on moving-average calculations. HMS argues that this forecasting approach struggles to produce accurate forecasts, particularly for products at the beginning or end of their life cycle. Although the method yields a decent overall forecast bias of approximately $\pm 10\%$ at category level, deviations at the individual item level are significantly larger. According to several studies, which will be discussed in this thesis, the implementation of ML-models for demand forecasting has been shown to produce more accurate results than traditional methods. HMS is therefore interested in investigating the possibility of conducting demand forecasting internally using ML-based approaches.

1.3 Problem Statement

Despite utilizing forecasting tools, such as Anaplan, HMS Networks experience limitations in their current demand forecasting. Traditional methods are efficient to identify long-term seasonalities and trends (Kumar Dubey et al., 2021). However, with unexpected variation of demand, due to e.g. large-volume orders, or a high turnover of products being introduced and phased out, such models are less reliable. This leads to insufficient forecasting precision, which in turn causes inventory levels to be either too low, increasing the risk of delays or lost sales, or too high, increasing costs and the risk of obsolescence (Petropoulos et al., 2022).

HMS Networks are in need of a systematic and data-driven forecasting method, which can handle both continuous and intermittent demand. ML-models have a greater ability to forecast such complex systems compared to traditional methods (Hammam et al., 2025). How the historical data is handled affects the performance of the ML-models. Therefore, to successfully implement the ML-models, a relevant aggregation of the demand data is required.

The challenge of the master thesis is to, by reviewing relevant literature, identify possible ML-models which can be applied to demand forecasting. Furthermore, these models should be developed and applied to the historical data of HMS Networks and analyzed based on performance. Thus, the thesis aims to contribute to both academia and the company by exploring the potential and practical implementation of machine learning as a tool for demand forecasting.

1.4 Purpose

The purpose of the master thesis is to explore how ML-models can forecast demand of a complex product portfolio, by applying the models to the case company data, with appropriate granularity, and using suitable metrics to compare the results to the current forecasting method used at the case company.

1.5 Research Questions

To achieve the purpose of the thesis, three research questions have been formulated.

1. *Which ML-models can be used to improve demand forecasting of a complex product portfolio?*

Research question 1 is posed to gain an understanding of how different types of ML-models perform in a demand forecasting scenario. The question is divided into two parts. Firstly, the review of literature and previous studies offers an overview of which ML-models have shown promising results when forecasting demand. Based on these findings the most promising ML-models are chosen and developed in the thesis. The second part of the research question is answered by applying the chosen ML-models to the case company and evaluating the performance, thereby answering which model/models achieve the best performance in the context of the case company.

2. *What levels of granularity (temporal and product hierarchy) offer the best conditions to generate a forecast on SKU level?*

Research question 2 is answered through the development and testing of the ML-models. It aims to understand which time intervals (e.g. weeks or months) and product aggregations (e.g. product groups or product categories) generate the most accurate forecasts. Since the choice of granularity and aggregation depends on the context of the forecast, research question 2 is case specific. Thereby, the answer applies to the scenario of HMS Networks rather than forecasting in general. However, a generalized discussion can be conducted based on the findings.

3. *How would a ML-based forecasting method perform compared to the current forecasting method used at HMS Networks?*

Research question 3 is divided into two parts. Firstly, the test scores of the ML-models are compared to the current forecast at HMS Networks to explore

if the ML-models can achieve more accurate results. Secondly, the research question is analyzed through a practical lens. The aim is to discuss whether an implementation of a machine learning-based forecasting method would be useful for HMS Networks in practice.

1.6 Delimitations

The thesis aims to understand the potential of machine learning within demand forecasting, as well as the potential of implementation at the case company, HMS Networks. Due to the conditions of the thesis, including limited time and resources, certain delimitations have been set to increase the feasibility of the project.

Firstly, the thesis is delimited to the process of demand forecasting, and does not include optimization of inventory policy (e.g. calculation of safety stock) or other operations despite them being indirectly affected by the demand forecasting process. This is due to the lack of data regarding inventory, as well as to maintain the focus on the process of demand forecasting.

Secondly, only historical sales data and product master data is used for the training of the ML-models. Thereby, the tests include a manageable amount of features and factors to analyze when developing the ML-models. The inclusion of external variables (e.g. macro factors) are not included in the tests due to lack of data and time limitations.

Thirdly, to ensure enough data to train and reliably evaluate the ML-models, only standard products in HMS Networks product portfolio are included in the thesis. Non standard products may be examined to deepen the understanding of the data but are not included in the scope of the thesis.

Lastly, the thesis is delimited to a proof-of-concept, evaluating the potential of machine learning as a tool for demand forecasting at HMS Networks. Although a practical implementation is discussed, a complete guide for IT-integration is not included in the scope of the thesis.

1.7 Disposition

The following list presents and describes the outline of the thesis.

2. Methodology	The chapter covers the research strategy and research process, introducing the Design Science Research methodology. The data gathering techniques, divided into review of literature and empirical data gathering, are described. Further, the data pre-processing steps are presented followed by a discussion regarding research quality and ethics.
3. Frame of Reference	The chapter is divided into three parts; demand forecasting, artificial intelligence and machine learning as well as machine learning in demand forecasting. Through presenting the literature findings of each part, a framework is created which describes the context and process of the thesis.
4. Empirics	The chapter describes the six pre-processing steps and the resulting data on which the ML-models are trained and tested. Further, the current demand forecasting process at HMS Networks is described.
5. Results	The chapter is divided into four parts; the final pre-processed data, the benchmarks, the results of the ML-models and the results of the current forecast at HMS Networks. The results of the ML-models are presented for each iteration of improvement.
6. Discussion	The chapter focuses on discussing the findings from the previous chapter. Firstly, the resulting data is discussed followed by the artifacts. The discussion of the artifacts is divided into the three cycles. Next, the results of varying granularity is analyzed followed by a comparison between the ML-models and the current forecast at HMS Networks.
7. Conclusion	The chapter circles back to the introduction by answering the three research questions based on the results and discussion. The contributions of the thesis are then presented. Lastly, limitations of the thesis are considered, including recommendations for future research.

2 Methodology

The methodology is the combination of approaches and data collection methods that together create the optimal conditions to achieve the purpose of the study (Björklund, 2012; Johannesson and Perjons, 2021). The choice of methodology is thereby based on the character of the research questions and the purpose of the study. In the following sections the research strategy, research process and gathering of research material are discussed. Thereafter, the various aspects of data pre-processing are explored. Lastly, research quality, research ethics and implementation details are discussed. The general structure of the methodology chapter is illustrated in Figure 1.

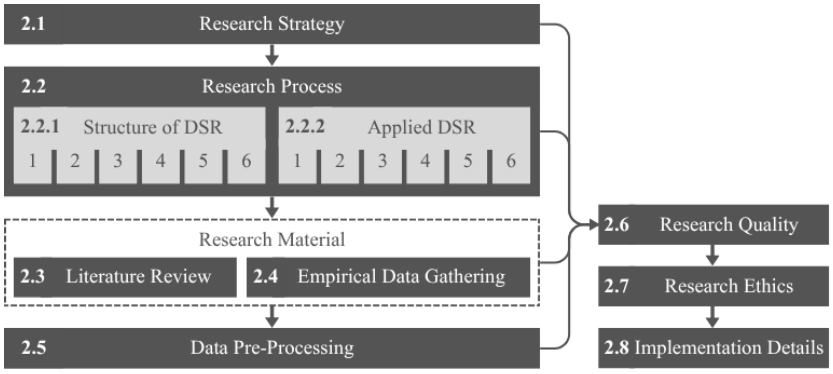


Figure 1: The structure of the methodology chapter.

2.1 Research Strategy

The overall plan of conducting a research study is defined as the research strategy (Johannesson and Perjons, 2021). Thus, research strategies offer high-level guidance for the researcher regarding planning, executing and monitoring the study. There is a variety of research strategies that can be applied depending on the structure of the thesis. Since this thesis attempts to develop a model which solves a specific research problem, Design Science Research (DSR) is deemed as an appropriate methodology. DSR is a constructive research methodology, with a focus on developing a design artifact to solve a concrete problem, while consequently applying the findings to answer knowledge questions (Knauss, 2021). Artifacts in DSR include e.g. constructs, models or methods that aim to improve upon a defined research problem (Peffers et al., 2007).

In this study the artifacts are ML-models that aim to improve the ability to forecast demand, which are evaluated by applying said models to historical demand and forecasting data from HMS Networks. The DSR methodology is therefore utilized to achieve full potential in the results of the models as well as the communication of results.

2.2 Research Process

As aforementioned, DSR was the research strategy utilized when conducting this thesis. The research process is a more detailed, step-by-step description of how to carry out the DSR process. These steps, as well as the application of the steps upon the thesis, are discussed in the following sections.

2.2.1 Structure of DSR

The process of DSR builds upon six steps (Peffers et al., 2007; Vom Brocke et al., 2020), as can be seen in Figure 2. By breaking down problems into, for example, goals and requirements, these steps enable designers to systematically approach problems and progress toward more specific solutions (Peffers et al., 2007).

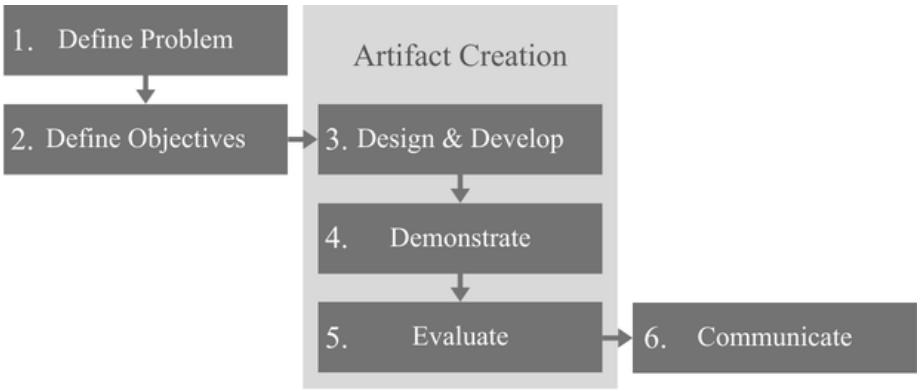


Figure 2: DSR Methodology inspired by Peffers et al., 2007, p. 54.

Step 1: Define Problem

The initial step of the process is to identify the specific research problem and to motivate the potential advantage of a solution. Vom Brocke et al. (2020) argue that motivating the value of the solution is essential as it incentivizes both the researchers and the audience to pursue the solution, while simultaneously demonstrating the researcher's comprehensive understanding of the problem (Vom Brocke et al., 2020).

Step 2: Define Objectives

The second step is to define the objectives for how the solution is expected to perform (Vom Brocke et al., 2020). These objectives should be derived from the problem definition, and constrained by what is technically feasible. They can be both quantitative, such as measurable performance improvements relative to current benchmarks, or qualitative, such as focusing on new capabilities or improved decision making support.

Step 3: Design & Develop

With the objectives set, the third step is the development of the artifact (Vom Brocke et al., 2020). This includes stating the desired functionality of the artifact and taking previous research into account when making decisions about the design of the artifact.

Step 4: Demonstrate

The fourth step of the DSR process is to demonstrate the use of the artifact to solve the stated problem (Vom Brocke et al., 2020). The demonstration should be done in an appropriate activity, such as using the artifact in a simulation or an experiment.

Step 5: Evaluate

The fifth step is to evaluate the artifact against the objectives set for the solution in step 2 (Vom Brocke et al., 2020). How this step is done, depends on the context of the problem and the set objectives. If, for example, there are quantitative objectives set for the solution, the artifact should be benchmarked against the set objectives in order to evaluate the performance of the designed solution.

Step 6: Communicate

The final step in the DSR process is communication, which involves passing on the research findings, the artifact's benefits and its contribution to the knowledge base. This step ensures that the insights gained from the design and evaluation of the artifact reaches all relevant stakeholders.

Despite being six individual steps, the DSR model should be interpreted as an iterative approach to research. If, during the evaluation or communication phases of the process, the artifact is deemed insufficient, Vom Brocke et al. (2020) suggest that the researchers may go back and redesign the artifact or develop a new artifact.

2.2.2 Applied DSR

When applying the DSR process to this thesis, certain alterations and specifications were made to tailor the process to the matter at hand. The application and alterations are discussed in the following paragraphs.

Step 1: Define Problem

The problem, and the possible value of a solution, were described in sections 1.3 and 1.4, respectively. HMS Networks has identified large forecasting biases, specifically at the SKU level. To establish an understanding of the context and potential solutions, previous literature as well as operations at HMS Networks

was studied. The frame of reference, seen in Chapter 3, covers three main areas; forecasting, machine learning, and machine learning in forecasting. Understanding these contexts allowed for a more supported approach for upcoming steps, regarding e.g. data collection and model choices.

Step 2: Define Objectives

The qualitative objective for the solution was to create an improved forecast for HMS Networks standard products. As this is a vague objective, different quantitative methods to evaluate forecasting performance were included in the frame of reference. Based on literature, certain metrics and benchmarks were selected to evaluate the research objective. Furthermore, information is gathered in this step through a review of literature, interviews and secondary data to understand the context in which the artifact exists.

Step 3: Design & Develop

The third step of DSR, the design and development phase, consisted of the practical building of the ML-models. Based on literature, relevant types of ML-models and data aggregations were concluded to be efficient for the given objectives. The raw data provided by HMS Networks were then pre-processed, and the relevant ML-models were built and trained using the historical data. Development of ML-models is in itself an iterative process, requiring an iterative experimental approach, since the model variables are gradually adjusted to achieve satisfactory results (Ranawana and Karunananda, 2021).

Step 4: Demonstrate

In step four, the models were demonstrated by testing them on unseen data to ensure accurate results. As this was a part of the iterative process, unsatisfactory results was solved by returning to step three and continuing the adjustments of variables. When satisfactory results were achieved, the models were deemed fully developed and the test results acted as the demonstration of the solution.

Step 5: Evaluate

Evaluating the solution consisted of comparing the performance of the artifacts against the accuracy of other models, benchmarks and HMS Networks current forecasting method. The metrics chosen were based on the literature reviewed in step 2. To further evaluate the artifacts on a more general level, they were compared to the benchmarks established during step 2. Thus, the evaluation of ML-model forecasts and the current forecasting results at HMS Networks was done using several different metrics and benchmarks. Furthermore, the evaluation included an analysis of the feasibility of an implementation of ML-models as forecasting tools.

Step 6: Communicate

The communication step was addressed through two primary channels, this report and a presentation at HMS Networks. The research process and empirical findings were documented in this report, while the presentation at HMS Networks aimed to ensure the utility of the artifact.

Updated DSR Methodology

When applying the DSR methodology to the thesis, the guidelines of Knauss (2021) were taken into account, guideline 2 having the largest impact on the research process. Guideline 2 suggests that the DSR process should be done in iterations, or so-called cycles (Knauss, 2021). In each cycle the artifact should be improved further, and knowledge regarding each research question should be increased. The recommendation of three cycles was adapted for the thesis. The updated DSR methodology including the three cycles is illustrated in Figure 3.

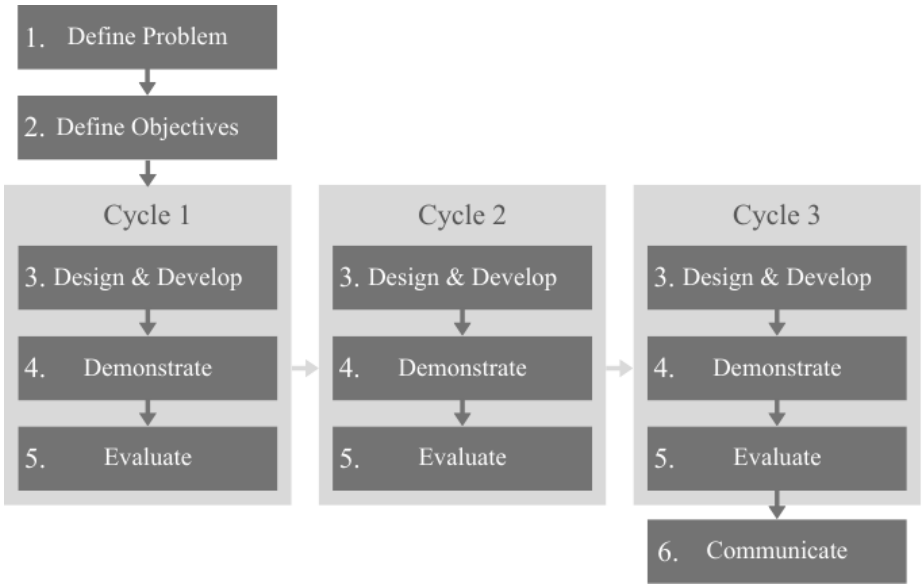


Figure 3: Updated Design Science Research Methodology.

For the first cycle, the main focus was gathering information and data. When the initial empirical data was collected, the first iterations of ML-models could be created. During each cycle the aim was to improve the artifacts, by analyzing the evaluation step from the previous cycle. This resulted in the second and third cycles focusing on further development of the models through restructuring the data for improved performance. During the second cycle, first order differencing (FOD) was applied. The third cycle consisted of improving the performance of the models by scaling the data. Both FOD and scaling is further discussed in Section 3.3.3. These cycles are thereby a three-time iteration of steps 3-5 in Figure 2.

2.3 Review of Literature

A literature review is the process of collecting information about the topic at hand through material such as books and academic journals (Björklund, 2012). The aim of a literature review is to summarize the most recent findings within the subject field (Rowley and Slack, 2004). A well constructed literature review facilitates identifying research topics, identifying research contributions, understanding theoretical concepts, choosing methodology and analyzing results. Furthermore, an advantage of literature reviews is the possibility to create a broad as well as deep frame of reference of the relevant subjects with limited time and resources (Björklund, 2012).

In this thesis, a review of literature was conducted rather than a fully systematic literature review. The purpose of this review was to establish a foundational understanding of key concepts and support the development of the study, rather than to exhaustively cover all existing research. As such, the review was more targeted and exploratory in nature. The review of literature conducted in this thesis focused on three subjects; demand forecasting, machine learning and machine learning applied in the field of demand forecasting.

The first subject, demand forecasting, was reviewed to understand the context and importance of demand forecasting. By understanding common practice within the field, potential challenges and opportunities that could affect the thesis were identified. Further, this review provided an understanding of common practices within the field regarding forecasting methods, metrics and benchmarks, which facilitated the analysis of the results. The second subject, machine learning, was studied to deepen the understanding of different machine learning models and their applicability for the thesis. Furthermore, the review of this subject provided the relevant knowledge to develop the artifact. The findings facilitated the process of improving the models, through concepts such as feature engineering. The last topic, machine learning in demand forecasting, provided an overview of historical research with topics similar to this thesis. By analyzing results of previous studies, a clearer course of action could be identified. Combining these three subjects created a broad knowledge base upon which the thesis could be executed.

Since a literature review is based on secondary data, it is important to question the information gathered (Björklund, 2012). Studies may be conducted for certain purposes, or be non-transparent of the applied methodology. It is therefore crucial to understand the context of the reviewed literature. To ensure reliability, the review of literature was, therefore, conducted with certain requirements. The literature should be peer reviewed, frequently cited and relevant. Thus, academic articles and books were used, with certain exceptions for books written by industry experts based on relevant experience. The literature search was executed mainly using Google Scholar and Finn (the database of Lund University's library). Key words used for the literature search were e.g. *demand forecasting*, *machine learning*, *artificial intelligence* and *design science research*. Rowley and Slack (2004) discuss different search strategies, whereof citation pearl growing and building blocks were applied in the literature review. Citation pearl growing is the process of gathering new key words from relevant sources to search for additional documents. Building blocks is the concept of extending the search to synonyms of key words to find a broader set of documents.

2.4 Empirical Data Gathering

The empirical data gathering consisted of conducting interviews with HMS Networks employees and gathering secondary data which constituted the basis of the artifact development. In the following sections, the approach of the empirical data gathering methods is described.

2.4.1 Interviews

Interviews are a method to gather primary data through direct contact with the interviewee (Björklund, 2012). Interviews allow for the gathering of specific information that is of interest for the thesis, but with the disadvantage of being highly time consuming (Johannesson and Perjons, 2021). The structure of an interview can be divided into three categories; structured interviews, semi-structured interviews and unstructured interviews (Björklund, 2012; Johannesson and Perjons, 2021).

A structured interview contains a predetermined list of questions, that are asked as written and in the specific order. This is particularly useful when conducting a large number of interviews, to ensure that all interviews are identically executed. A semi-structured interview is based on predetermined questions or topics, but allows for more variation in formulation and follow-up questions based on the respondents answers. An unstructured interview is when the questions are formulated during the interview, and thereby creates more of an open discussion with the interviewee. This is useful when the aim is to create a general understanding of a topic.

In this thesis, a combination of semi-structured and an unstructured interview have been conducted. The interviewees were employees at HMS Networks, in-

cluding executives, supply planners and a market manager. These interviews offered insights of the respective areas of expertise of the interviewees. The semi-structured interview focused on market overview, customer behavior and promotions. The questions of the semi-structure interview are found in Appendix A. The unstructured interviews offered a general understanding of HMS Networks as a company and the challenges faced within the demand forecasting process.

The majority of the interviews were conducted through Microsoft Teams, due to geographical distance. During the unstructured interviews one of the authors took notes. The semi-structured interview was recorded and could thus be transcribed after the interviews were completed. A summary of the conducted interviews and the topics they addressed is shown in Table 1.

Position	**Topic**	**Date**
Chief Financial Officer	Current practice and challenges of demand forecasting at HMS Networks	2025-11-07
Chief Operating Officer	Current practice and challenges of demand forecasting at HMS Networks	2025-11-07
Senior Supply Planner	SIOP process at HMS Networks	2026-02-25
Market Manager	Market strategy and presence of promotions at HMS Networks	2026-03-09
Director Business Control Supply Chain	Weekly meetings about progress and data	Weekly meetings (Jan-May 2026)

Table 1: List of conducted interviews.

2.4.2 Secondary Data

Secondary data, or documents (Johannesson and Perjons, 2021), is information that has already been collected for other purposes. Secondary data includes several different types of media, such as texts, images, raw data or videos. To distinguish between the collected secondary data and the review of literature, which is also based on secondary information, the secondary data was defined as internal documents from HMS Networks. Thus, the secondary data consists of spreadsheets and presentations created by HMS Networks.

In this thesis, the secondary data collected from HMS Networks were relevant for the development and evaluation of the artifacts. The documents consisted of sales data, product data, forecasts and presentations. The sales data included

the orders for each product during the period 2021-2025. The sales data were the foundation for the development of the artifacts. The product data contained the multiple levels of product categorization, which allowed for data aggregation in the artifacts. HMS Networks' forecasts for 2025 were collected, to function as a benchmark for the evaluation of the artifacts' performance. Finally, presentations regarding the sales, inventory and operations planning (SIOP) within HMS Networks were acquired to deepen the understanding of the context of the thesis. A list of the secondary data acquired from HMS Networks is presented in Table 2.

Document	**File Type**	**Description**	**Application**
Sales data	Excel	All orders during 2021–2025	Artifacts & benchmarks
Product data	Excel	Master data including product categorization and life-cycle status	Aggregation & filtering
SIOP deck	PowerPoint	Description of the SIOP process within HMS Networks	SKU dis-aggregation
HMS Networks Forecast	Excel	12-month forecast for 2025	Benchmarks

Table 2: List of documents acquired from HMS Networks.

2.5 Data Pre-Processing

Collecting raw data is not enough to achieve valuable insights. To be able to draw any conclusions, the data must be analyzed and transformed into meaningful and manageable information (Johannesson and Perjons, 2021). Since the ML-models were trained solely on historical sales data, it was crucial to clean up and structure the data to enable relevant and comparable results. In this thesis, the data pre-processing consisted of five main steps:

1. Filtering
2. Time clumping and zero-fill
3. Grouping
4. Data splitting
5. Stock-outs and promotions

Firstly, filtering was the process of excluding products that exceeded the scope of the thesis. Secondly, time clumping and zero-fill included restructuring the sales data into a time series format and filling empty time intervals with zeros. Thirdly, grouping consisted of aggregating products from SKU level to product groups and categories by mapping the sales data to the product data. Fourthly, data splitting was the process of dividing the time series data into training data, validation data and test data, in accordance to the practice of training and validating ML-models. Lastly, the presence of both stock-outs and promotions were investigated and taken into account if necessary. Further description of the four pre-processing steps is found in Section 4.1.

2.6 Research Quality

The trustworthiness of a study can be assessed through three primary criteria; validity, reliability, and objectivity (Björklund, 2012). Within the context of DSR, these concepts are closely related to the rigor and relevance of the developed artifact and the evaluation process (Hevner et al., 2004). A study should aim to achieve high validity, reliability, and objectivity simultaneously, as these dimensions collectively determine the overall rigor of the research. The three criteria, as well as measures that can be taken to ensure them, are discussed in the following sections.

2.6.1 Validity

Firstly, validity refers to the extent to which the study measures what it intends to measure (Björklund, 2012), and whether the designed artifact effectively addresses the identified problem. In DSR, this is often ensured through appropriate evaluation methods, such as experiments, simulations, or case studies, which demonstrate that the artifact performs as intended in its application environment (Peffers et al., 2007). The validity of the thesis is achieved by training all ML-models on the same training data, validating with the same validation data and testing it on the same unseen test data. Furthermore, several metrics are used to evaluate the models for further validity. Lastly, the models are compared to both benchmarks and the forecast created by HMS Networks, using several metrics.

2.6.2 Reliability

Secondly, reliability concerns the repeatability of the research process and its outcomes. In an engineering context, this implies that the design, implementation, and evaluation procedures are sufficiently documented to allow other researchers to replicate the study or achieve comparable results under similar conditions. Transparent modeling choices, clearly defined parameters, and reproducible computational procedures are essential to achieving high reliability (Hevner et al., 2004). In this thesis, each step of the process is thoroughly documented with transparency, with the aim of increasing the reliability. This

includes the pre-processing and structure of the data as well as the development of the ML-models, by describing each development cycle. Furthermore, the use of metrics and benchmarks is described to further increase the reliability and repeatability of the thesis.

2.6.3 Objectivity

Thirdly, objectivity refers to the extent to which the research is conducted without bias and is not unduly influenced by the researcher's subjective judgments. In DSR, complete objectivity can be challenging, as the researcher is actively involved in artifact creation. However, objectivity can be strengthened through systematic evaluation, the use of quantitative performance metrics, and comparison against baseline methods. Additionally, clearly articulating assumptions and limitations helps mitigate the influence of researcher bias (Hevner et al., 2004; Peffers et al., 2007). To remain objective, the results were evaluated using quantitative and standardized methods including metrics and benchmarks, which were selected prior to any results. Furthermore, all artifacts were generated with identical conditions, to ensure objectivity. The expected outcomes were discussed based on previous studies within the field. However, the ML-models were evaluated solely based on the performance metrics, avoiding any influence from the assumptions on the results.

2.7 Research Ethics

Despite being a commonly applied research methodology, the description of DSR rarely includes ethical considerations (Myers and Venable, 2014). Therefore, Myers and Venable (2014) have derived a set of six ethical principles for design science research, shown in the following list.

1. The public interest
2. Informed consent
3. Privacy
4. Honesty and Accuracy
5. Property
6. Quality of the artifact

The first principle, public interest, focuses on the affects the developed artifact may have on the safety, health, democracy, empowerment and emancipation of the public. This principle, though important, is not particularly relevant for this thesis. Developing ML-models for internal demand forecasting is not considered a risk to the public.

The second principle, informed consent, concerns the consent of any individual who is included in the thesis. The individuals included in this thesis are interviewees. To follow the second principle, all interviewees have given consent

to be included in the thesis. Furthermore, all data included in the thesis has used with consent from the case company, thereby avoiding the inclusion of confidential information.

The third principle, privacy, is connected to the second principle and considers the privacy of individuals in the thesis and the artifact. The privacy of interviewees is taken into consideration by omitting any names, and only including job titles in the thesis. The principle of privacy in the artifact is irrelevant since the ML-models do not process any personal information.

The fourth principle, honesty and accuracy, focuses on honesty in the presentation of results as well as not plagiarizing other sources. This principle is followed by clearly presenting all results before analyzing and discussing them. Plagiarism is avoided by acknowledging and stating all sources that have inspired decisions in the thesis.

The fifth principle, property, concerns the ownership of the artifact and the results. The written report is property of the authors, while the artifact and results of the report is property of HMS Networks. This means that HMS Networks have the right to use the artifact and results in a potential implementation.

The sixth and last principle, quality of the artifact, focuses on the development and safety of the artifact. This principle is followed by ensuring that each iteration of the artifact development phase aims to improve the artifact. The safety aspect of this principle is not relevant for this thesis, as no safety is put at risk during the development of the ML-models.

2.8 Implementation Details

When conducting the thesis a number of programs and tools were used to facilitate the process. These tools were applied to several aspects of the project, from developing the artifacts to writing the thesis. A list of the tools and their respective application is seen in Table 3. Furthermore, the code was written using the programming language Python version 3.13.7.

Area of Use	**Tool/Program**	**Application**
Writing	LateX	Writing and structuring the thesis
	ChatGPT	Wording and spell-check
Coding	Visual Studio Code	Development environment for models and data processing
	GPT-5.3-Codex	Code generation and debugging
	Claude Sonnet 4.6	Code generation and debugging
	GitHub	Version control of code
	Optuna	Hyperparameter tuning
	React	Framework used to build UI for artifact development
Literature searching	Google Scholar	Literature search
	Finn	Literature search
	Zotero	Reference management
Data handling	Microsoft Excel	Structuring and analyzing data
Meetings & Interviews	Microsoft Teams	Conducting online interviews and meetings

Table 3: Tools used in the study.

3 Frame of Reference

The frame of reference of the report presents the relevant theoretical knowledge that is applied in the thesis. The chapter is divided into three parts. Firstly, an overview of demand forecasting is presented, which sets the historical context of forecasting practices, as well as the aspects that must be understood and considered when attempting to forecast demand. Secondly, the concept of machine learning is presented, as well as the relevant subcategories and models that are used in the thesis. Lastly, the two previous subjects are combined by presenting literature which discusses the implementation of machine learning as a demand forecasting tool. This serves the purpose of showing different approaches, as well as the potential and limitations of different models. The general structure of the frame of reference chapter is illustrated in Figure 4.

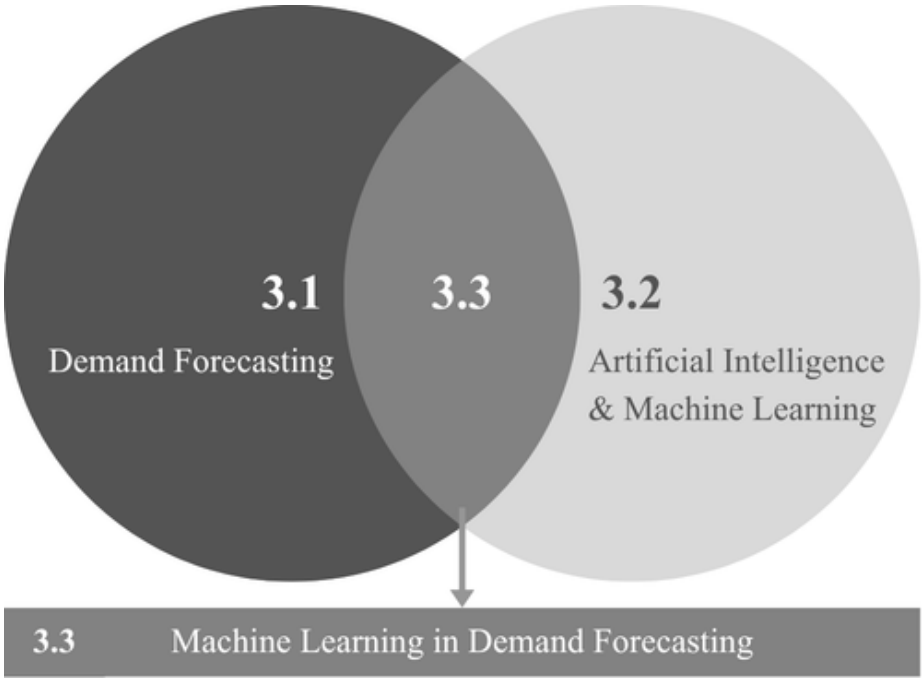


Figure 4: Illustration of the frame of reference chapter.

3.1 Demand Forecasting

The demand forecasting section covers relevant knowledge for developing demand forecasting models. This includes; understanding the data, different forecasting methods, relevant metrics and benchmarks. The following sections aim to create an understanding of the demand forecasting process that will act as a foundation for the development of the artifacts.

3.1.1 The Purpose of Demand Forecasting in SCM

Demand forecasting represents a critical component of supply chain management (SCM) (Ingle et al., 2021). Supply chains rely on making appropriate decisions concerning a wide variety of operational and strategic planning activities. A substantial proportion of supply chain activities concern uncertain future events (Petropoulos et al., 2022; Kolassa et al., 2023). The question that lies at the core of these decisions is how much demand is expected in the future (Petropoulos et al., 2022). To answer this question, demand forecasting provides relevant and meaningful information, consequently supporting supply chain decision making (Ingle et al., 2021; Petropoulos et al., 2022; Kolassa et al., 2023). Vandeput (2023) simply states that the better the demand can be estimated, the better the decisions. Kolassa (2023) emphasizes an important distinction is that the forecast itself is not the same as a target, plan or decision. A forecast is a prediction of the future which should, if accurate, facilitate business decisions. The primary purpose of forecasting is therefore not precision, but uncertainty reduction and risk mitigation (Petropoulos et al., 2022).

The benefits of effective forecasting are well documented. Accurate forecasts contribute to lower inventory levels, reduced stock-outs, smoother production plans, reduced operational costs, and improved customer service (Ingle et al., 2021; Abolghasemi et al., 2020). Consequently, forecasting accuracy is not merely a statistical objective, but a driver of organizational performance.

Forecasting is grounded in the assumption that historical observations contain information relevant for predicting future outcomes (Petropoulos et al., 2022). However, accurate demand forecasting is a challenging endeavor. The underlying volatility in demand causes difficulties in performing demand forecasting with a high level of precision (Abolghasemi et al., 2020; Petropoulos et al., 2022). Decisions based on forecasts that turn out to be inaccurate can cause unnecessary costs in procurement and transportation, manpower, service level, and inventory. To support the choice of an optimal forecasting method and reduce uncertainty, it is necessary to understand the demand characteristics.

3.1.2 Demand Data Characteristics

Forecasting literature consistently emphasizes that no single model universally outperforms all alternatives across different contexts (Abolghasemi et al., 2020). Model performance is highly dependent on demand characteristics. Traditional statistical models (which are discussed further in Section 3.1.3) such as ARIMA and exponential smoothing have been shown to outperform artificial neural networks under stable demand conditions, whereas machine learning approaches may yield superior results in volatile environments (Abolghasemi et al., 2020). This underscores the importance of contextual model selection, and understanding the demand data. In the upcoming sections relevant aspects of demand data characteristics are presented.

Time Series Analysis

To understand the characteristics of demand data, it is necessary to understand the structure of time series data. Time series is a sequence of similar measurements that have been taken at regular intervals (Kolassa et al., 2023). All time series include a time index, and at least one numerical variable that represents the observation at each time index.

Time series analysis is the concept of analyzing historical time series data to obtain information about the future (Kolassa et al., 2023). Demand forecasting is thus a type of time series forecasting, which uses historical data to predict future behavior of the time series. To practice time series forecasting, the historical data must be bucketed into periods. The most common period to forecast is monthly, but can also be weekly, daily, hourly or yearly depending on the process. The frequency of observation, and other parameters of the time series, directly affects the choice of forecasting model (Petropoulos et al., 2022), which will be further discussed in the upcoming section about aggregation and granularity.

Demand Uncertainty and Forecastability

Demand uncertainty constitutes a central challenge in forecasting. Volatile demand patterns may generate significant operational inefficiencies if uncertainties are not properly incorporated into planning decisions (Abolghasemi et al., 2020). While increasing inventory buffers or capacity levels may mitigate the negative consequences of demand variability, such approaches are often costly and inefficient (Abolghasemi et al., 2020). Due to volatility of demand data, forecasting cannot be expected to create a precise prediction (Petropoulos et al., 2022). Rather, the result of a forecast can be divided into an expected value (point of forecast), a prediction interval, a percentile and an entire prediction distribution. Thus, the uncertainty of the forecast and different scenarios can be considered. However, for time series with a high level of unexplainable variation, there is a limit to how well it can be forecast (Kolassa et al., 2023). In particular, low-volume and intermittent demand processes tend to exhibit reduced predictability due to sparse observations and irregular purchasing patterns. Understanding demand variability and forecastability is essential for selecting appropriate forecasting methodologies.

A specific factor which may affect the forecastability of demand is the presence of promotions and other strong drivers that affect demand (Abolghasemi et al., 2020; Vandeput, 2023). This includes e.g. campaigns, discounts or events that have an impact on the demand before, during or after the promotions (Abolghasemi et al., 2020). As a result, the volatility of the demand series increases, and the forecastability is reduced. It is therefore necessary to examine the presence of promotions to improve the performance of forecasting models. The case of promotions at HMS Networks is further discussed in Section 4.1.6.

Aggregation and Granularity

With unlimited resources, time and data, forecasts could, in theory, be computed to the highest possible granularity (Petropoulos et al., 2022), e.g. each SKU for every hour. However, in practice the resources are limited, meaning that data must be aggregated. Both temporal granularity (e.g., daily, weekly, monthly) and cross-sectional granularity (e.g., SKU, product group, region) influence forecasting performance and applicability (Kolassa et al., 2023). Forecast design requires careful consideration of aggregation levels and granularity. While higher levels of aggregation often improve forecast accuracy through the cancellation of random fluctuations, they may obscure information necessary for operational decision-making (Kolassa et al., 2023; Petropoulos et al., 2022). For example, forecasting large product categories can have excellent accuracy, Furthermore, highly granular forecasts may capture detailed demand patterns but are frequently associated with higher noise-to-signal ratios (Petropoulos et al., 2022). Forecasts should therefore be generated at aggregation levels aligned with the decisions they support (Vandeput, 2023).

Comparison of Demand and Sales Data

A critical distinction in forecasting concerns the selection of the forecast variable. A demand forecast should be based on demand data, which is what, when and how much a customer wants (Vandeput, 2023). Retrieving demand data is, however, incredibly challenging. Common practice is to track sales data instead of demand, thus creating a forecast based on sales data. Forecasting sales rather than demand may introduce structural distortions, particularly in the presence of stock-outs (Kolassa et al., 2023; Petropoulos et al., 2022; Vandeput, 2023). Sales are inherently constrained by supply availability, whereas demand represents the unconstrained customer requirement (Vandeput, 2023). If the stock-levels are high enough, all demand will be met, and correspond to the sales data. However, during stock-outs the sales data will be zero, while demand may still exist. A forecast based on sales data will therefore interpret the demand as zero, and forecast accordingly. In the most extreme case, the forecast will be zero, and the stock-out will continue. Since the sales will consequently remain at zero, the forecast will be determined as accurate, although the actual demand is ignored.

Reliance on sales data alone may therefore produce misleading forecasts. For instance, stock-out conditions may artificially depress sales observations, potentially leading to perpetuated under-forecasting (Petropoulos et al., 2022; Kolassa et al., 2023). To mitigate such biases, it is necessary to account for inventory shortages and track periods of constrained sales (Petropoulos et al., 2022). The presence of stock-outs at HMS Networks is further discussed in Section 4.1.6.

3.1.3 Common Forecasting Methods

To understand the context of which this thesis is written, a few common forecasting methods are briefly explained. Forecasting methods can broadly be divided into simple baseline approaches and more advanced statistical models.

Simple Forecasting Methods

Simple forecasts, such as the historical mean, moving average, naive forecast, and seasonal naive, are easy to implement and often surprisingly effective, in stable environments with limited external drivers (Kolassa et al., 2023). The historical mean assumes demand fluctuates randomly around a constant level, and forecasts the demand by averaging the historical values. Moving average (MA) is similar to the historical mean, with the key difference of only averaging a certain number of the most recent historical values. For example, in a monthly time series, a six month moving average (6M MA) forecasts the average of the previous six values. The naive forecast projects the most recent observation forward, offering high adaptability but also higher variance. Seasonal naive methods extend this idea by incorporating repeating seasonal patterns. Although these simple methods perform poorly when applied to time series with higher variability, they can be valuable as components in ensemble forecasting, where multiple forecasting methods are combined, often improving accuracy over individual models (Kolassa et al., 2023). They can also be used as benchmarks, to facilitate the evaluation of other methods, which will be discussed further in Section 3.1.5.

Statistical Forecasting Methods

More advanced methods include exponential smoothing, Holt-Winters approaches and ARIMA models. Exponential smoothing is a more complex version of MA. It applies decreasing weights to older observations, balancing responsiveness and stability through a smoothing parameter. It can be seen as a compromise between moving averages and naive forecasts and remains widely used due to its simplicity and effectiveness (Ingle et al., 2021; Kolassa et al., 2023). The Holt-Winters model extends exponential smoothing by incorporating both trend and seasonality, allowing for additive or multiplicative effects (Ingle et al., 2021). ARIMA (AutoRegressive Integrated Moving Average) models generalize both autoregressive (AR) and MA approaches. These models capture linear dependencies in time series data, where current values depend on past observations and past errors. The integration component models changes in the series rather than levels, making ARIMA suitable for non-stationary data (Ingle et al., 2021; Kolassa et al., 2023). Although theoretically more flexible than exponential smoothing, ARIMA models do not always outperform simpler methods in practice (Kolassa et al., 2023).

3.1.4 Forecasting Metrics

To be able to determine the quality of a forecast, it is important to measure the deviation between the forecast values and the actual values. However, there are several different ways to interpret this information. Different metrics have different outcomes and lead to different insights (Vandeput, 2023). In other words, there is no predetermined way to evaluate if a forecast is *good* or *bad*. Rather, it depends on which metric or metrics that are chosen, emphasizing the importance of choosing correctly (Mehdiyev et al., 2016). Vandeput (2023) states that there is no one-size fits all approach when it comes to forecasting metrics. Different metrics have varying focus on extreme values, bias and horizons. However, all metrics have in common that they are based on the forecast error (e_t), defined as the difference between the forecast value (f_t) and the actual value (d_t), for the time t, as seen in Equation 1.

$$e_t = f_t - d_t \tag{1}$$

A positive error is an indicator of over-forecasting, while a negative error is an indicator of under-forecasting (Vandeput, 2023). Keeping track of over- and under-forecasting is the difference between the two main ways of evaluating forecasting errors, accuracy and bias. Accuracy looks at the average magnitude of error, while bias keeps track of if there is a trend of over- or under-forecasting. The goal is to create an accurate and unbiased forecast, which highlights the need to keep track of both accuracy and bias through different metrics (Vandeput, 2023). An illustration of the difference between accuracy and bias is seen in Figure 5.

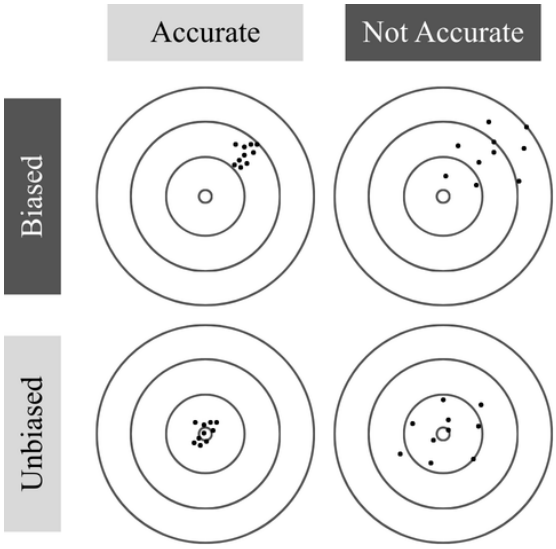


Figure 5: Illustration of accuracy and bias inspired by Vandeput, 2023, p. 62.

Bias

Bias is determined by calculating the average error of each time t over an interval of n periods, as seen in Equation 2.

$$Bias = \frac{1}{n} \sum_{t=1}^{n} e_t \tag{2}$$

Bias simply tells us the average error, which in turn is a good indicator to identify structural issues in the forecast, causing general over or under-forecasting (Vandeput, 2023). However, using bias as the only evaluation metric risks overlooking important information. If some periods have very positive errors, while others have extremely negative errors, the bias can be near zero even though the forecast is *bad*. To be able to identify this issue, accuracy should be measured alongside bias.

Accuracy

Compared to bias, which has a straight forward definition and calculation, accuracy is more complicated to define. There are several different ways to calculate accuracy, all of which have different advantages and disadvantages. Different accuracy metrics are better suited for certain scenarios, and less suited for others. In this thesis, three of the most commonly used metrics for accuracy will be discussed; mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean squared error (RMSE). These error measures are widely used due to their recognized relevance and simplicity (Voyant et al., 2025).

The first accuracy metric, MAE, is calculated similarly to bias, with the key difference of averaging the absolute error rather than the error (Vandeput, 2023), as seen in Equation 3.

$$MAE = \frac{1}{n} \sum_{t=1}^{n} |e_t| \tag{3}$$

The metric thereby determines the average magnitude of the errors for a certain time interval, without weighting or ignoring certain values (Liapis et al., 2021). MAE is simple to understand and implement, but does not consider bias whatsoever. If a forecast is optimized based on minimizing MAE, the forecast tends to move towards the median demand, rather than the mean (Vandeput, 2023). This happens if there are occurrences of extreme values that increase the mean. Imagine a time series with three periods and the values 1, 1 and 13. The mean is 5 and the median is 1. If we would forecast 5 for each period, the MAE would be 5.33, while a forecast of 1 would have a MAE of 4. Furthermore, since MAE provides a balanced measure of average forecast error, extreme deviations that may heavily affect the planing process process are not emphasized (Voyant et al., 2025).

The second accuracy metric, MAPE, is the most commonly used metric to measure forecasting accuracy, according to Vandeput (2023). MAPE is calculated by first looking at each period individually, and determining the absolute percentage error (APE) for each period, as seen in Equation 4.

$$APE_t = \frac{|e_t|}{d_t} \tag{4}$$

The average of APEs over the forecast time interval corresponds to the value of MAPE, as seen in Equation 5.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n} APE_t = \frac{1}{n}\sum_{t=1}^{n} \frac{|e_t|}{d_t} \tag{5}$$

Vandeput (2023) continues by stating that MAPE is a flawed metric, since the percentages are calculated for each data point. This results in an automatic weighting of errors, where an error is weighted more if the demand is low. For example, a constant forecasting error of 5 units will generate a higher APE-value during low demand. This results in MAPE generally under-forecasting, since the metric reacts better when the forecast is catered towards the low demand periods instead of to the high. The largest issues for MAPE occur when the demand or forecast is 0, since the APE is either 0 or undefined (Liapis et al., 2021). When handling intermittent demand, MAPE does not offer sufficient insights. Although, an advantage of MAPE compared to MAE and RMSE, is that MAPE is not scale-dependent, thereby allowing comparisons between different time series (Liapis et al., 2021).

The final accuracy metric, RMSE, is the most complex of the three. It is calculated by squaring the error, and taking the square root of the average of all squared errors, as seen in Equation 6.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2} \tag{6}$$

RMSE is more complicated to compute, as well as explain, compared to MAE and MAPE, but generates relevant insights (Vandeput, 2023). What differentiates RMSE is the fact that the errors are squared, thereby emphasizing large errors and comparatively negating small errors. This method represents the cost of forecasting errors quite well, but issues arise when sudden outliers occur (Liapis et al., 2021; Vandeput, 2023). A forecast which performs reliably under normal conditions may receive a poor RMSE score due to infrequent but significant forecast errors (Voyant et al., 2025).

Choice of Metric

As previously mentioned, no metric is perfect. MAE skews the forecast towards the median, RMSE may be over-affected by outliers an MAPE often causes under-forecasting and handles intermittent demand poorly. A model which performs well based on RMSE may perform poorly when measured in MAE, due to the sensitivity to large errors (Voyant et al., 2025). Furthermore, it is insufficient to solely measure accuracy, and thereby ignore the affect of forecasting bias. Vandeput (2023) therefore recommends a combination of metrics that considers both aspects. By equally weighting each error unit, MAE is more neutral than MAPE and RMSE. By combining MAE and bias, a metric that combines simplicity and robustness is created. Note that the absolute value of bias is used, as seen in Equation 7, to avoid rewarding negative bias.

$$MAE + |Bias| = \frac{1}{n}\sum_{t=1}^{n}|e_t| + |\frac{1}{n}\sum_{t=1}^{n}e_t| \tag{7}$$

The combination of metrics seen in Equation 7 is the main metric considered when comparing the artifacts. However, since all the aforementioned metrics provide different insights, all of them are included in the thesis for later analysis. Notable is that MAE and RMSE are scale-dependent (Liapis et al., 2021).

Although choosing a suitable set of metrics is an efficient way to quantify the quality of a forecast, it is useless if there is nothing to compare the forecast to. This is where the concept of benchmarking becomes relevant.

3.1.5 Forecasting Benchmarks

When measuring the accuracy and bias of a forecast, it is relevant to compare it to a point of reference. Common practice is to e.g. compare to last year's forecast, predetermined accuracy targets or visual assessment (Vandeput, 2023). Although these methods offer some relevant insights, they have their weaknesses. Comparing to last year's forecast can be misleading since external factors can cause forecasts to perform better or worse certain years. Comparing to an accuracy target is a reasonable idea, though it is difficult to determine what the target should be. Visual assessment of the forecasts is straightforward, but difficult to scale when handling a large number of SKUs or time periods. A more neutral way to assess forecasting metrics is to use benchmarks. These benchmarks are simple forecasts, that are often insufficient as stand-alone forecasts, but can contribute by being used as baselines or sanity checks (Kolassa et al., 2023). Three common forecasts that are often used as benchmarks are naive forecasts, historical mean and moving average.

Naive forecasts are commonly used as benchmarks, but have high inaccuracy when applied to demand with high variability (Vandeput, 2023). The model is based on simply forecasting the demand value from the previous time period.

The historical mean is calculated by averaging the demand over the entire historical time interval. It is surprisingly effective when forecasting demand with low variation or trends, but creates large errors when handling demand with any significant noise (Kolassa et al., 2023). Compared to the previous benchmarks, MA is a bit more complex. Similarly to the historical mean, MA is based on averaging historical values (Vandeput, 2023). However, instead of averaging all historical values, each MA forecast is the average of a certain number the most recent time periods. Vandeput (2023) argues that 3-month and 6-month moving averages often perform well, by considering historical variability without including too old data.

In Figure 6 the benchmarks; historical mean, naive forecast and 6-month moving average are visualized in a graph using randomized values as the demand. As previously described, the naive forecast follows the real values with a delay of one month. The historical mean averages out with time, similarly to the 6M MA, with the difference of 6M MA being slightly better at following recent trends. This is, as previously mentioned, due to the fact that the 6M MA does not include unnecessarily old data, compared to the historical mean. Worth noting is that the benchmarks forecast one month forward, compared to forecasts that commonly forecast a longer time interval.

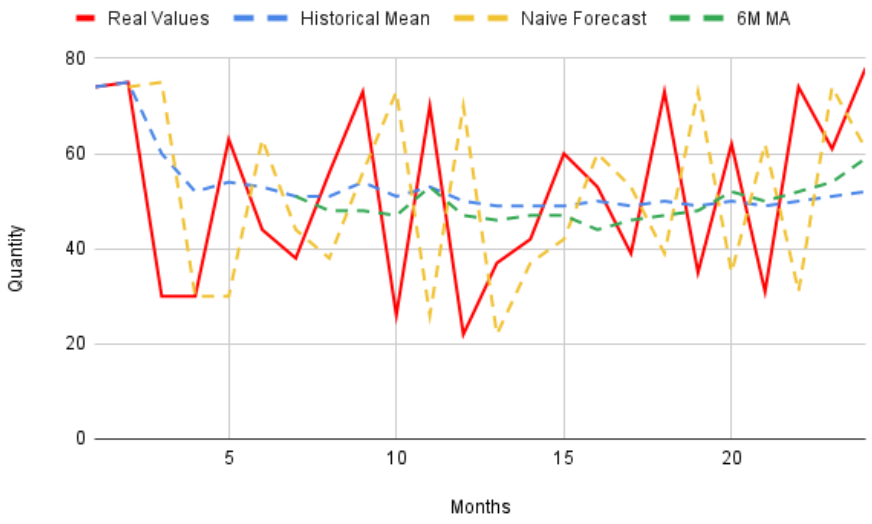


Figure 6: Visualisation of benchmarks; historical mean, naive forecast and 6-month moving average.

3.2 Artificial Intelligence and Machine learning

The term artificial intelligence (AI) was introduced by Marvin Minsky and John McCarthy in 1956 (Haenlein and Kaplan, 2019). It refers to a broad field of computer science and engineering that aims to develop systems capable of solving real world problems through intelligent behavior (Ertel and Mast, 2025). The earliest studies including AI revolved around the thought that machines would become intelligent if they understood logical reasoning (Zhou, 2021). The initial results were promising, but people started to realize that allowing machines to solely use logic reasoning was not enough to enable artificial intelligence. Within the field of AI, machine learning has emerged as one of the most prominent tools used to allow machines to learn knowledge through experience. While AI refers broadly to the concept of making systems capable of intelligent behaviour, ML is a tool which aims to create algorithms that learn patterns directly from data rather than relying on explicitly programmed rules.

Machine learning algorithms are commonly divided into two main categories based on the type of training data, supervised and unsupervised learning (Nasteski, 2017; Zhou, 2021). Supervised learning refers to when the model is trained on input attributes alongside an expected output attribute (Berry et al., 2020). The objective of the model is to find patterns between the input and output attributes to correctly predict the output solely based on input attributes. Unsupervised learning refers to models trained on data without predefined output attributes. These models aim to find hidden patterns and structures within the data.

Within supervised learning, algorithms can be further divided by the type of prediction they are designed to produce (Berry et al., 2020). The two main categories of supervised learning are classification models and regression models. The objective of a classification model is to assign observations to a finite set of predefined classes. Regression models, on the other hand, focus on predicting continuous numerical values from observations. The choice between classification and regression models therefore depends on the characteristics of the objective. Since demand is a continuous numerical quantity, demand forecasting naturally falls within the scope of supervised regression problems. A simplified illustration of the categorization within ML is seen in Figure 7.

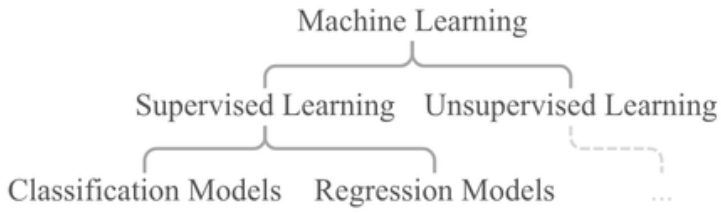


Figure 7: Simplified overview of top-level categorization within machine learning.

3.2.1 Regression in Machine Learning

Regression models aim to model and predict a dependent variable, y, from a set of covariates, $x_i, ..., x_n$ (Fahrmeir et al., 2022). A defining characteristic of regression is that the relationship between the dependent variable and the covariates is not a deterministic function, but subject to random variation. Consequently, the dependent variable is treated as a random variable with a distribution that depends on the covariates. Even though the covariates are known, the dependent variable can not be known, but rather its expected value and variation can be estimated. Therefore, regression models aim to characterize the conditional expectation of the dependent variable, y, given the known covariates, $x_i \dots x_n$, as seen in Equation 8, where f is the regression function.

$$E(y|x_i, \dots, x_n) = f(x_i, \dots, x_n) \tag{8}$$

Since the regression function provides only an estimate of the expected value of the dependent variable, an error term is introduced to account for deviations between the modeled relationship and the observed value, as seen in Equation 9. The value, f, is called the systematic component and the error, e, is called the stochastic component.

$$y = E(y|x_i, \dots, x_n) + e = f(x_i, \dots, x_n) + e \tag{9}$$

In practice, the systematic component is unknown and must be approximated from data. Since the systematic component is unknown, different regression models attempt to approximate this function using different modeling strategies. These models vary in terms of their assumptions, structure and learning procedures, leading to different strengths and limitations.

In this thesis, the subcategory of regression models explored are called tree-based models. This choice is motivated by the findings presented in Section 3.3.6. The name tree-based refers to the general structure of the models, which resembles an upside-down tree. Within the category of tree-based models, weak predictors and ensemble models are of interest. The following sections introduce the concept of four ML-models, which constitute the artifacts developed in this thesis. The four models are; Decision Tree, Random Forest, LightGBM and XGBoost. An extended version of Figure 7, including further categorization and the ML-models of interest, is shown in Figure 8.

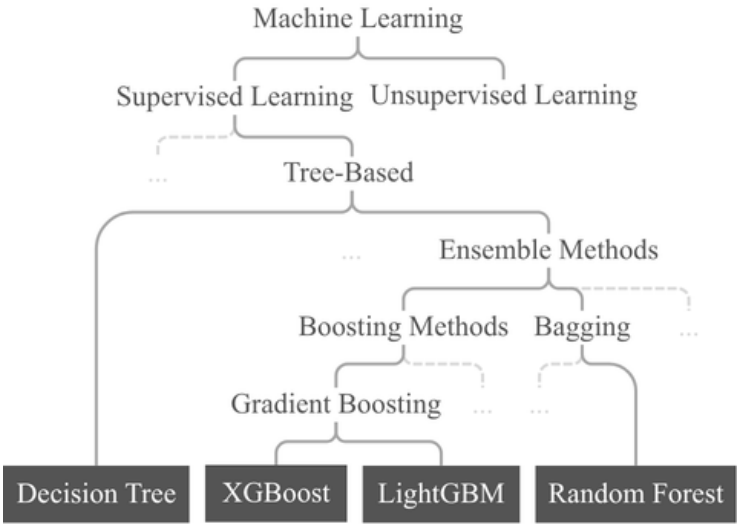


Figure 8: Simplified overview of tree-based supervised learning methods used in the thesis.

3.2.2 Decision Tree

One approach to approximate the systematic component, f, is through the use of a Decision Tree (De Ville, 2013). There are two main types of Decision Trees, classification trees which are used when the dependent variable is discrete and regression trees used for continuous dependent variables. They work by recursively partitioning the input space into smaller regions based on the value of the covariates. The resulting model can be visualized as a hierarchical tree structure, as seen in Figure 9, consisting of a root node, internal decision nodes, branches, and terminal nodes. The top node, usually referred to as the root node, contains the full dataset and represents the overall distribution of the dependent variable. Each internal decision node further separates the observations into smaller regions. The estimate of the systematic component is produced in the terminal nodes of the tree, often corresponding to the average of all the estimates within the same region (Fahrmeir et al., 2022).

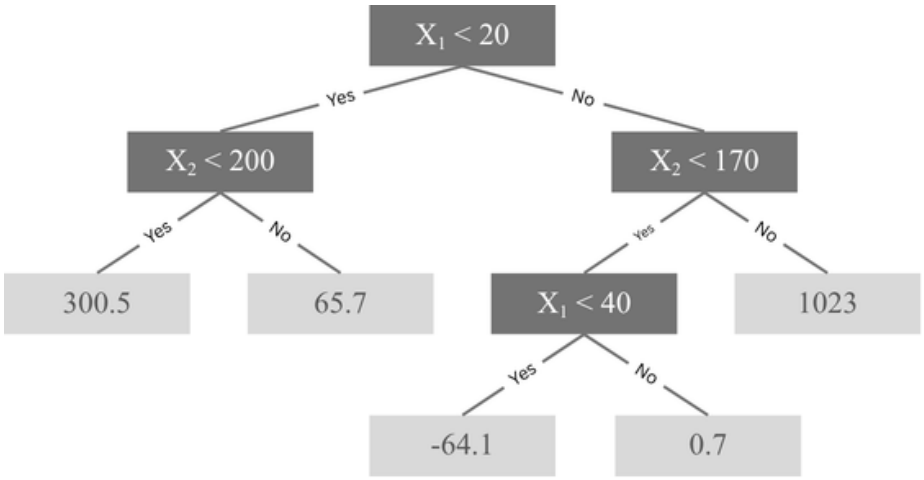


Figure 9: Example of a regression tree structure, inspired by Henckaerts et al., 2019, p. 6.

The growth of a Decision Tree starts at the root node, where the algorithm evaluates potential splits based on the available covariates (De Ville, 2013; Fahrmeir et al., 2022). Each split divides the data according to a category, threshold or interval of a selected covariate, with the aim to create regions containing observations that are more homogeneous with respect to the dependent variable. To determine which split is the best, the algorithm evaluates a set of possible splits and optimizes a predefined loss function or impurity measure (Fahrmeir et al., 2022). These measures quantify the degree of heterogeneity within a node, allowing the algorithm to select the covariate and corresponding split point that yields the greatest improvement in each node (De Ville, 2013). Once a split has been decided, the same process is repeated recursively for each resulting subset of observations, progressively growing the tree structure. The recursion continues until a predefined stopping criterion is reached.

When predicting a continuous dependent variable, the impurity measure is usually defined in terms of the variability of the observations within a given node (Kuhn and Johnson, 2013). A node is considered purer than its parent node when the variance in the dependent variable is less. In regression trees, this decrease in heterogeneity is commonly evaluated using measures such as sum squared error or mean squared error, which quantify how closely the predicted value within a region represents the observed data. Consequently, the optimal split is the one that produces nodes with an overall lower prediction error compared to the parent node.

Although Decision Trees provide an intuitive approach to approximate the regression function, single-tree models show several limitations, making them suboptimal in certain situations (Hastie et al., 2009; Kuhn and Johnson, 2013). A

major concern for single tree models is their high variance (Hastie et al., 2009). Even small changes in the data can give rise to very different sequences of splits, and hence very different predictions. This is caused by the sequential structure of the tree, where errors in the root split are propagated to all the splits below it. Another limitation, specific to regression trees, is that the predictions are piecewise constant rather than smooth, which may limit their ability to predict a smooth relationship.

A key limitation of tree-based models in general is their lack of extrapolation capability (Hatakeyama-Sato and Oyaizu, 2021). Since predictions are derived from piecewise constant regions defined by the training data, these models do not learn an explicit functional relationship between inputs and outputs. Consequently, when applied to trending time series, they tend to produce flat or bounded predictions within the observed data range, making them unsuitable for capturing trends.

3.2.3 Random Forest

Ensemble based approaches were originally developed to reduce variance and to improve the predictive accuracy by combining multiple predictors into a single model (Zhang and Ma, 2012). In the context of Decision Trees, this approach is implemented by aggregating the predictions of multiple trees into a single ensemble model. One of the most widely used ensemble techniques is the Random Forest algorithm, which creates multiple trees, all intentionally different from each other. The final estimated prediction of the model is an aggregation of all predictions from the set of Decision Trees in the ensemble, illustrated in Figure 10.

The construction of a Random Forest includes randomness at various levels, used to lower the correlation between individual trees (Zhang and Ma, 2012). Each tree is trained on a bootstrap sample of the data, obtained by randomly drawing observations with replacement, from the entire dataset. The method of having multiple predictors trained on bootstrap samples of the data is commonly referred to as bagging or bootstrap aggregation. The result is an increased diversity among trees. In addition to this, the algorithm also introduces randomization in the splitting procedure by only considering a randomly selected subset of covariates at each individual node. By restricting the candidate variates for splitting, the correlation between trees is lowered. Each tree in the ensemble is grown independently using the same recursive partitioning procedure as regular Decision Trees, which is shown in Figure 9. The final prediction of the Random Forest, in the context of regression, is commonly obtained by an unweighted average of the individual predictions of all the trees in the ensemble.

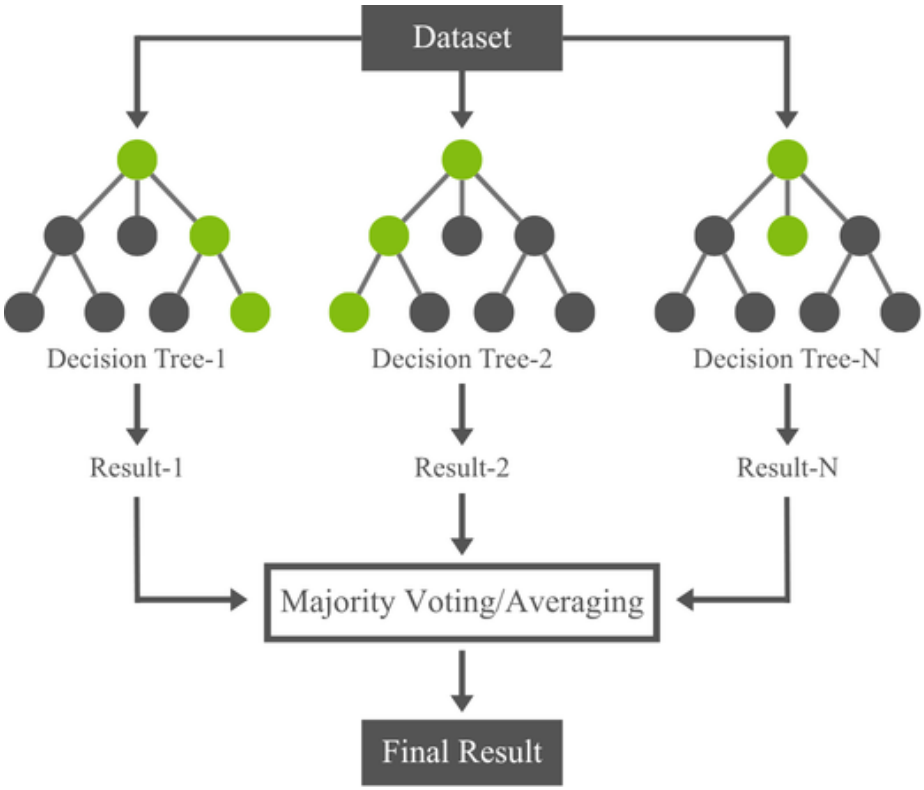


Figure 10: Example of a Random Forest structure, inspired by Gunay, 2023.

Compared to single Decision Trees, Random Forests offer several advantages. By combining multiple trees in an ensemble, the model reduces variance and improves predictive stability, which is one of the central motivations to use ensemble models (Zhang and Ma, 2012). Because the individual trees are trained on bootstrap samples and with randomized covariate subsets, every tree in the ensemble becomes less correlated, allowing the aggregation step to produce more reliable estimates (Breiman, 2001). However, these improvements come at the cost of reduced interpretability, since the structure of ensemble models makes it more difficult to trace individual decision paths compared to regular Decision Trees. Another disadvantage is that the resulting regression function remains piecewise constant and therefore lacks smoothness.

3.2.4 Boosting

An alternative method to construct ensemble models is boosting (Zhang and Ma, 2012). In contrast to bagging based methods, boosting builds models sequentially instead of in parallel (Duffy and Helmbold, 2002). Each new predictor is built with an increased focus on observations that were poorly estimated by

previous predictors, allowing the ensemble to iteratively improve its predictive performance (Zhang and Ma, 2012). Between iterations, the training data is modified such that mispredicted observations receive greater focus in the training of subsequent estimators.

An extension of the boosting framework is gradient boosting, where the predictive model is improved by sequentially adding new learners that reduce the remaining model prediction error of the ensemble. While the traditional boosting framework increases focus on the poorly predicted observations, gradient boosting accomplishes this by directly modeling the prediction errors of the ensemble (Friedman, 2001). The model is therefore updated in a step-wise manner, where each additional predictor provides an incremental correction to the regression model by minimizing a predefined loss function. This model structure is illustrated in Figure 11, where r_N corresponds to the residuals from tree N, $\hat{y}_N$ and $\hat{r}_N$ to the predictions generated by tree N.

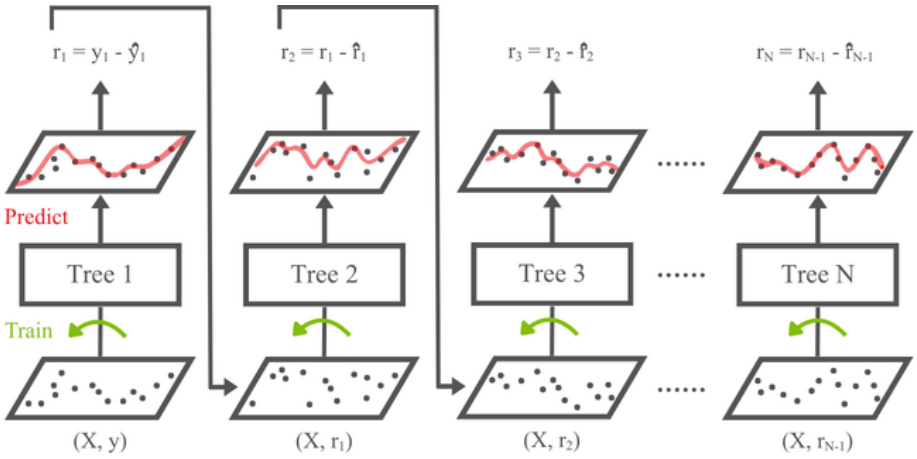


Figure 11: Structure of Gradient Boosting model, inspired by Özkurt, 2024, p. 9.

3.2.5 LightGBM

Light Gradient Boosting Machine (LightGBM) is an implementation of the gradient boosting framework designed for high efficiency and scalability (Ke et al., 2017). However, instead of growing each sequential tree level wise, it uses a node wise growth algorithm, which is illustrated in Figure 12. At every step, it adds the node with the greatest information gain. The resulting models often become deeper and more complex, which can improve the models predictive accuracy. However, the main advantages of LightGBM become evident mostly on high dimensional datasets. Many of the innovations used in the implementation are aimed at improving computational efficiency while maintaining predictive accuracy. In addition, the leaf wise growth of the model may lead to more

complex trees, which can increase the risk of overfitting if not properly tuned, introduced in Section 3.2.7 and Section 3.2.8 (Ke et al., 2017).

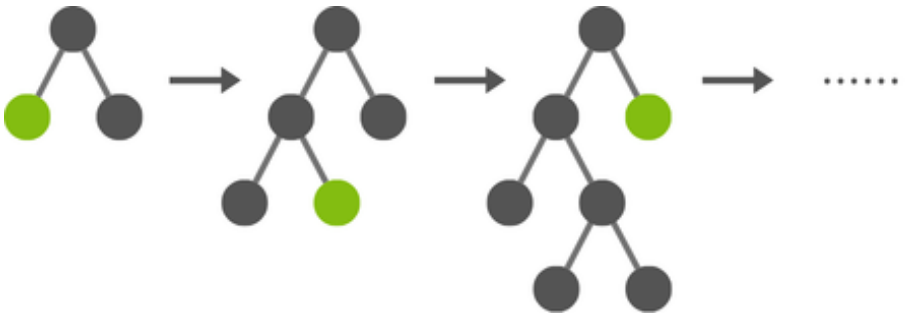


Figure 12: Leaf wise tree growth of LightGBM model, inspired by Bouhamed et al., 2022, p. 7.

3.2.6 XGBoost

Extreme Gradient Boosting, commonly called XGBoost, further extends the concept of gradient boosting. The general structure of the model remains sequential, but the objective function incorporates an additional term that penalizes model complexity (Chen and Guestrin, 2016). The objective function combines a predefined loss term, with a regularization component that constrains the structure of the trees. Instead of fitting trees directly to the target value, XGBoost constructs each new tree using gradient based optimization, where both first- and second-order information of the loss function are used to evaluate potential splits. This information allows the algorithm to determine not only how predictions should change, but also how strongly each update should be applied. In addition, XGBoost grows trees in a level wise manner, resulting in a more balanced tree structure as illustrated in Figure 13. In contrast to LightGBM, XGBoost does not incorporate the same computational optimizations, which may result in a higher computational cost. Additionally, like in the case of LightGBM, its performance depends strongly on the choice of hyperparameters, requiring careful tuning (see Section 3.2.7).

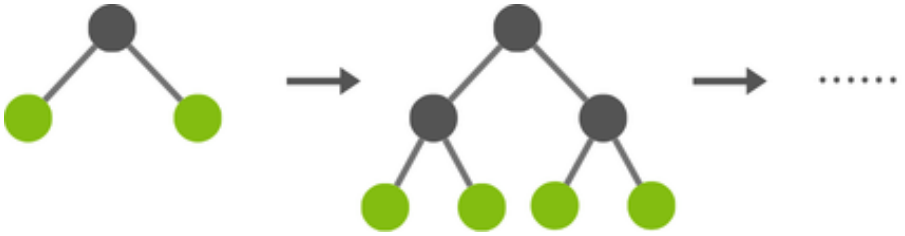


Figure 13: Level wise growth of XGBoost model, inspired by Bouhamed et al., 2022, p. 7.

3.2.7 Hyperparameters

Almost all machine learning models rely on model parameters and hyperparameters (Zhou, 2021). While model parameters refer to the parameters learned by the model itself during training, hyperparameters are usually set in advance. Different choices of hyperparameters can lead to models with significantly different performance. Hyperparameters control key aspects of the models learning process, such as the complexity and model structure. As a result, they can greatly affect the models' ability to learn and generalize from the training data.

Different models have different sets of hyperparameters. For example, in Decision Trees, hyperparameters such as maximum depth determine how complex the final model can become, while Random Forest models are structured by the number of trees (Zhou, 2021). Boosting models, such as LightGBM and XGBoost, have hyperparameters that control the number of sequential iterations (Ke et al., 2017).

Parameter Tuning

The procedure of finding a fitting set of hyperparameters is commonly referred to as parameter tuning (Zhou, 2021). As mentioned, hyperparameters are not learned during training, but instead specified in advance. Therefore, this process involves evaluating multiple models trained with different hyperparameter configurations and then selecting the configuration that performs best according to a predefined evaluation metric.

In practice, it is not possible to try all combinations within the parameter domain, as they can be real-valued (Zhou, 2021). Hence, the true optimal configuration is unlikely to be found. Instead, parameter tuning typically involves searching through a limited subset of the parameter domain. This introduces a trade-off between computational cost and model performance, as evaluating a larger number of configurations may increase the likelihood of finding a better performing model, but also increase the required training time. Since each configuration corresponds to training and evaluating a new model, the total training time grows close to linearly with the number of configurations.

3.2.8 Overfitting

When training ML-models, it is important to ensure that the model performs well not only on the training data, but also on the unseen data. Although the goal of an ML-model is to minimize a loss function, a loss value very close to zero in the training data is not necessarily good (Zhou, 2021). Such results may indicate that the model has fitted the training data too closely, capturing noise or specific patterns instead of the underlying relationships. This phenomenon is commonly referred to as overfitting. In contrast, a model that performs well on both the training data and unseen data is said to generalize well.

The risk of overfitting increases with the complexity of the model, as they are more capable of capturing noise in the training data (Zhou, 2021). Training a model with a limited number of data points is another common reason for overfitting (Ying, 2019), as the model has fewer observations from which to learn the underlying patterns. Within time series forecasting, the amount of data is often restricted by the observation period and level of aggregation. This may further increase the risk of overfitting.

In contrast to overfitting, underfitting is a phenomenon that commonly occurs when a model is too simple to reliably capture the patterns in the data (Aliferis and Simon, 2024). A common example of underfitting is the use of a linear model to approximate a non-linear relationship. As a result, the model performs poorly on both the training data and the unseen data. One possible cause of this is the restrictive hyperparameters, such as Decision Tree depth, creating models that are too simple. Another potential cause is insufficient features, which may prevent the model from identifying meaningful patterns within the data.

3.2.9 Data Splits

In order to evaluate the hyperparameters and to guide the model selection, the available data is typically divided into training, validation and test data sets (Hastie et al., 2009). The training set is used to fit the model, while the validation set is used to tune and evaluate hyperparameters. The test data is hence only used to evaluate the final model. The distinction between validation and test data is important in order to ensure an unbiased estimate of the models generalization performance. If the test data were to be used during hyperparameter tuning, the model could become fitted to the test data, and lead to overly optimistic performance estimates.

In the context of time series data, the process of dividing the data set into sections is less straightforward due to the time dependency (Hyndman and Athanasopoulos, 2014). Since observations are temporally ordered, random splitting can lead to information leakage, where information about the future is used to predict the past. Furthermore, the characteristics of the time series data may differ over time, meaning that different periods may exhibit various levels of volatility and/or trend.

As a result, a simple fixed validation period may not be representative of the entire data set (Hyndman and Athanasopoulos, 2014). To address this, time series specific validation strategies such as time series cross-validation approaches can be implemented. These methods preserve the temporal order of the data, while providing a better estimate of model performance across various time periods. An example of how time series cross-validation is structured is illustrated in Figure 14.

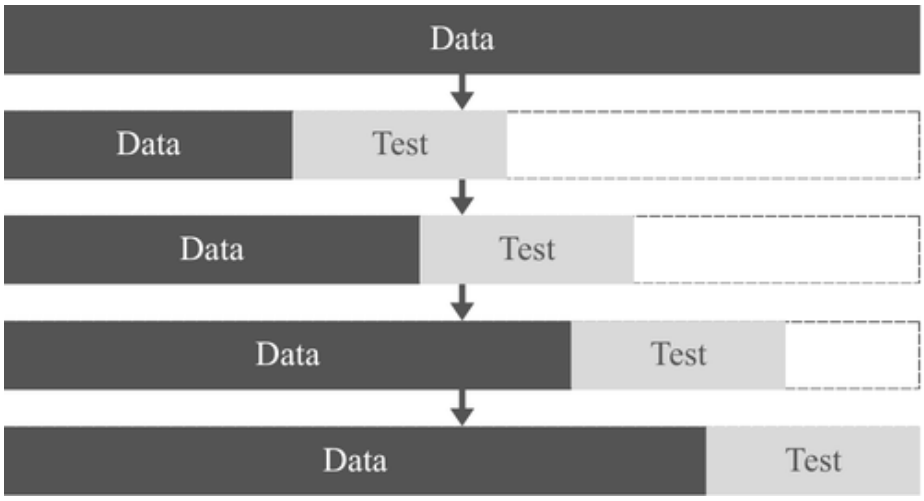


Figure 14: Example image of how time series cross-validation works, inspired by Shrivastava, 2020.

3.2.10 Comparison of Models

The models considered differ in terms of complexity and structure, but they all belong to the same class of tree-based models. These models share a common approach to regression by partitioning the feature space into regions with similar target values. This allows them to capture non-linear relationships and interactions between variables, given an appropriate data representation. However, due to their structure, they are limited in their ability to extrapolate beyond the observed data. As a result, they may struggle to generalize beyond the range of previously observed values.

The models address these limitations in different ways. Random Forest reduces variance by constructing multiple trees from different subsets of the data, leading to more robust and stable forecasts. The boosting based models, LightGBM and XGBoost, instead build the trees sequentially, which allows them to capture more complex patterns and achieve reduced bias. The strengths and weaknesses of the models are summarized in Table 4.

ML-model	Strengths	Weaknesses
Decision Tree	Simple and interpretable	Instability and possibly too simple
Random Forest	Robust, reduces variance, and provides stable predictions	Less interpretable and limited smoothness
LightGBM	Efficient, fast, and able to capture complex patterns	Sensitive to noise and requires tuning
XGBoost	Captures complex patterns and includes strong regularization	Computationally heavier and requires tuning

Table 4: Comparison of the ML-models used in the thesis, based on the review of literature.

3.3 Machine Learning in Demand Forecasting

In sections 3.1 and 3.2, demand forecasting and ML have been discussed. In this section, the combination of these two subjects will be explored, as illustrated in Figure 4. The focus is on discussing factors that affect the potential of ML in demand forecasting, as well as present findings from previous attempts in the literature.

3.3.1 Context of ML in Demand Forecasting

Babai et al. (2023) have studied the number of articles published within the field ML in SCM. The systematic review showed substantial increase of articles since 2019. Furthermore, the largest subcategory of articles focused on ML in supply chain planning, with a majority of articles specifically discussing demand forecasting (Babai et al., 2023). Thus, ML algorithms have gained significant attention within the field of demand forecasting in recent years (Abolghasemi et al., 2020), especially in cases with a high mix and low volume (HMLV) product mix (Wang, 2025). The growing complexity of modern markets contributes to this methodological shift. Continuous product innovation and shortened product life cycles generate increasingly diverse demand patterns, thereby increasing the need for accurate and adaptive forecasting methods (Hammam et al., 2025).

Despite the increasing popularity of ML solutions, it remains challenging to draw general conclusions regarding their performance relative to traditional statistical methods, as results vary depending on context and data characteristics (Abolghasemi et al., 2020). However, since demand is influenced by random, uncertain, and sometimes fuzzy factors, establishing precise mathematical representations using purely linear assumptions is often difficult (Hammam et al.,

2025). In this context, machine learning, as a subfield of artificial intelligence, enables predictions based on historical data without being constrained by strict linear assumptions. Consequently, the ability of ML-models to detect patterns and trends in historical and external data positions them as valuable tools for effective supply chain management (Nagadevi et al., 2025).

ML enhances the ability of firms to automate forecasting across multiple time series and incorporate diverse internal and external drivers (Leokhin et al., 2025; Nagadevi et al., 2025). Improved forecasting accuracy supports more informed decision-making in SCM, particularly regarding safety stock determination, production planning, and service-level optimization (Nagadevi et al., 2025).

Given the increasing complexity of product life cycles and nonlinear demand patterns, ML shows great potential within the field of demand forecasting. Notable is that it should be regarded as an extension of traditional statistical approaches rather than a universal replacement (Abolghasemi et al., 2020; Hammam et al., 2025). Model selection should, therefore, be guided by demand characteristics, forecasting horizon, and empirical validation.

3.3.2 Feature Selection

Within all fields of machine learning, the concept of feature selection is important and can be time consuming. The features, which correspond to the covariates from Section 3.2.1, are the explanatory variables of the model. The choice of features can hence greatly affect the ability of the model to learn from the data (Zhou, 2021). By excluding important features, the models predictive ability might become reduced due to the information loss. On the contrary, including irrelevant features may increase the model complexity and the risk of overfitting.

In the context of time series forecasting, historical observations are commonly used as features (Hyndman and Athanasopoulos, 2014). By including previous observations as lagged values, the model is able to capture temporal dependencies such as trends and seasonal patterns in the data. However, feature selection is a complex task that often requires iterative testing and evaluation. Other methods to improve the effectiveness of a given set of features exist, such as feature engineering.

3.3.3 Feature Engineering

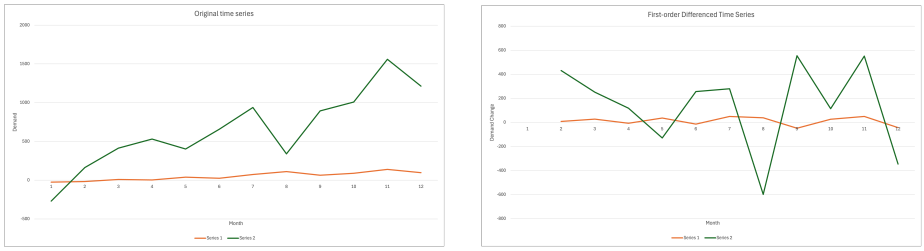
Feature engineering refers to the process of transforming raw data into another format that can be utilized effectively by machine learning models (Verdonck et al., 2024). Rather than relying solely on the original data, additional features can be constructed to better represent the patterns in the data. This is particularly important in the context of time series forecasting, where temporal data must be represented in a way that allows the model to learn the relevant structures, such as seasonality and trends.

An example of feature engineering in time series data is the transformation of timestamps into informative features. For instance, a timestamp on the form $DD/MM/YYYY$ can be decomposed into values of e.g. week, month or year allowing the model to capture recurring seasonal patterns. Similarly, lagged target values and rolling means can be introduced to provide the model with information about past observations (Verdonck et al., 2024).

In addition to constructing new features, feature engineering can also be used to address limitations of the underlying model. As discussed in Section 3.2.2, tree-based models have a limited ability to extrapolate estimates outside of the range of the training data, which can make trending time series data difficult to predict. Feature engineering can therefore be applied to transform the data so that the target value becomes more suitable for the given model, through detrending.

Detrending Transformations

One widely used approach to detrend time series data is to use first order differencing (Altunkaynak and Nigussie, 2018), which removes the trend component by extracting the difference between consecutive observations. As a result, the transformed time series usually becomes more stable in mean, which can make patterns other than trend easier to model. An example of FOD applied to two time series is illustrated in Figure 15. One limitation of FOD, which may be more pronounced for short series, is that it reduces the number of data points in the series by one.



(a) Original Time Series (b) First-order Differenced Time Series

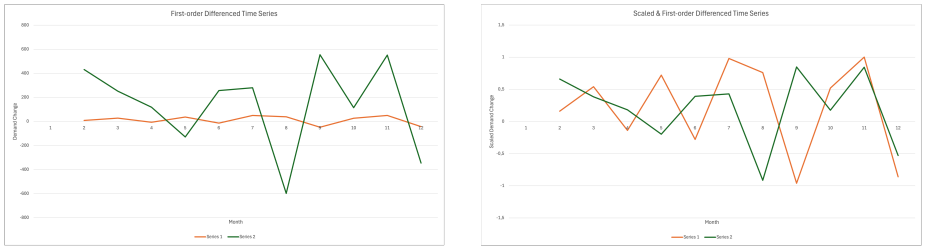
Figure 15: First-order Differencing applied on example data.

Other approaches to detrend time series data include explicitly modeling and removing the trend component, for example using linear regression (Altunkaynak and Nigussie, 2018). However, many of these methods rely on assumptions about the form of the trend, which may not hold in the future as demand patterns change. These methods are therefore not considered further in this study. These transformations are particularly relevant when using tree-based models, as they transform the problem, reducing the reliance of extrapolation.

Scaling

Another aspect that may be relevant is the scale of the observations, especially when the individual time series exhibit different magnitudes (Montero-Manso and Hyndman, 2021). When multiple time series are used to train a single model, large differences in scale can affect the learning process, as series with greater magnitude may have a greater influence on the model.

Another benefit of using scaling, in combination with a global model, is that it can enable cross-series learning (Montero-Manso and Hyndman, 2021). By transforming each series to a comparable scale, the model may become better at identifying and learning shared patterns across different series. The process of scaling can be performed in several ways, typically by normalizing the series by a characteristic value such as the mean or standard deviation (Montero-Manso and Hyndman, 2021). An example of scaling being applied to two time series with varying magnitudes is illustrated in Figure 16.



(a) FOD Time Series

(b) Scaled & FOD Time Series

Figure 16: Scaling applied on example data.

3.3.4 Forecasting Methodologies

When creating forecasts that span over multiple time steps into the future, it is necessary to determine how these future time steps should be handled. This problem is commonly referred to as multi-step forecasting, and different strategies can be applied depending on how predictions are produced for each horizon (Taieb and Hyndman, 2012).

The traditional multi-step forecasting methodology is the recursive method (Taieb and Hyndman, 2012). In this method, a single model is trained to predict one time step ahead. This model is then applied repeatedly to generate forecasts further into the future. The previously predicted value is inserted in order to generate the following prediction. This recursive pattern is then followed until the desired forecasting horizon is reached, as illustrated in Figure 17. This approach is both computationally efficient and simple, as only one model is required. However, a limitation of this method is that prediction errors accumulate over time, since each forecast depends on previously predicted values rather than actual observed ones.

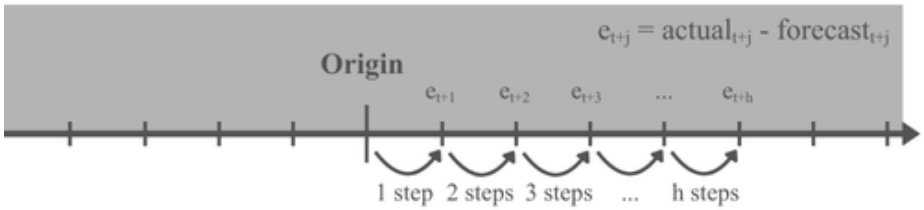


Figure 17: Illustration of recursive forecasting strategy, inspired by Svetunkov, 2024.

An alternative approach to the recursive method is the direct forecasting strategy (Taieb and Hyndman, 2012). In this approach, separate models are trained for each step in the forecasting horizon. When making predictions for future values, all models are then used to retrieve the total horizon forecast. For a twelve month forecast horizon, with one month time steps, this approach incorporates 12 individual models, as illustrated in Figure 18. Unlike the recursive approach, the direct approach does not rely on previous predictions as input. Instead, each forecast is generated independently based on observed data. This eliminates the issue of errors that accumulate over time. However, the direct approach requires training multiple models, which may require large amounts of data to ensure that each time step in the forecast horizon is reliable.

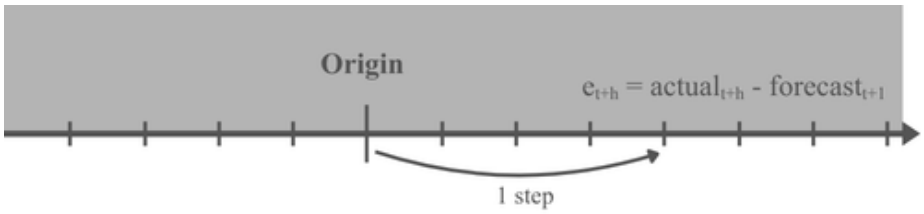


Figure 18: Illustration of direct forecasting strategy, inspired by Svetunkov, 2024.

In addition to the recursive and direct approaches, there are other forecasting strategies and hybrid methods, which combine elements of both approaches. However, these are not considered further in this study.

3.3.5 Model Structure

In the context of forecasting the demand for multiple product groups/categories, it is also necessary to determine whether a single model should be trained across all series or if separate models should be constructed for each individual group or category. These approaches are commonly referred to as global and local modeling strategies (Montero-Manso and Hyndman, 2021).

In a local approach, a unique model is trained for each individual time series, allowing each model to focus exclusively on the characteristics of a single series (Montero-Manso and Hyndman, 2021). However, this approach introduces several limitations. When the available data for each series is limited, the resulting models may become unreliable and prone to overfitting. Furthermore, since each model is trained independently, it is not able to find potential relationships or shared patterns between different product groups/categories.

In contrast, in the global modeling approach, a single model is trained using all series simultaneously (Montero-Manso and Hyndman, 2021). In this setting, product specific data such as product group or category must therefore be included as an input feature in order for the model to differentiate between the data points. This enables the model to learn shared patterns across multiple series, such as common trends or seasonal effects.

As a result, global models may generalize better, particularly in settings where individual series are short or sparse (Montero-Manso and Hyndman, 2021). However, local models may still be advantageous when individual series exhibit distinct behavior, as they allow for more specialized modeling of each series. Hence, the choice between a global or a local approach becomes a trade off between leveraging shared information and capturing series specific characteristics, and is often limited by the amount of data available for each individual series.

3.3.6 Previous Studies

Leokhin et al. (2025) studies the performance of six different ML-models (Linear regression, Random Forest, XGBoost, LightGBM, CatBoost and LSTM). The study is carried out through three separate experiments, each having different datasets with different horizons and product and time aggregations. Furthermore, three different metrics were used to evaluate the performance of the models (RMSE, MAE and R2). All three experiments showed that tree-based models yielded the best results, while LSTM had low performance. Specifically, XGBoost and LightGBM consequently performed well, and showed great potential after hyperparameter tuning. However, Leokhin (2025) emphasizes that the risk of overfitting must be considered when applying these models.

Nagadevi et al. (2025) applied XGBoost to forecast demand. XGBoost was compared to three other forecasting methods (SARIMA, Decision Trees and Random Forest). When applying the same metrics as in the study by Leokhin et al. (2025), XGBoost generated superior results according to all metrics ((Nagadevi et al., 2025).

As mentioned in Section 3.3.1, the performance of ML-models compared to statistical forecasting methods varies depending on the linearity of the data. To tackle this problem, Hammam et al. (2025) developed a hybrid model. The concept was to first apply the ARIMA model, then check for nonlinearity. In the cases of nonlinearity, XGBoost was incorporated to address the complexities of

the data. The study showed that for data with high variability, the hybrid model showed substantial improvements compared to traditional ARIMA (Hammam et al., 2025).

Fouad et al. (2025) conducted a study comparing four different ML-models (XGBoost, LSTM, Random Forest and Gaussian Regression) to ARIMA in a demand forecasting context. The results showed that all ML-models slightly outperformed the traditional ARIMA model. LSTM achieved the best performance, followed by XGBoost (Fouad et al., 2025).

Wang (2025) developed a hybrid model, Dynamic Dual-Phase Forecasting Framework (DDPFF), to more accurately forecast cases with a lack of historical data. This includes cases of HMLV and cold-starts, i.e. newly introduced products that therefore lack data. Such cases are famously difficult to forecast using traditional statistical models. The DDPFF model approaches the issue by combining ML and statistical models, applying XGBoost, weighted regression and ARMA in succession. DDPFF outperformed traditional ARIMA as well as an analogous forecast based on all applied metrics (Wang, 2025).

Wandre et al. (2025) forecast the demand of a pharmaceutical supply chain using three different methods; XGBoost, Random Forest and Prophet (an open source additive regression model developed by Meta). Naive forecasts and ARIMA were used as benchmarks, and several metrics were applied. The results showed that XGBoost outperformed the other models, followed closely by Random Forest on the short term forecasts (Wandre et al., 2025).

The literature review of studies that have explored ML as a tool for demand forecasting consistently shows three main themes. Firstly, the studies show that ML-models perform well, and in many cases outperform traditional statistical models. Secondly, the subcategory of ML that generates the greatest results is tree-based models. Lastly, in the studies where the XGBoost model is included, it is among the best, if not the best, performing model. In studies with hybrid models, XGBoost is included as the ML segment, further depicting its potential in the field of demand forecasting. Furthermore, Random Forest and LightGBM show promising results in several studies.

4 Empirics

The empirics chapter is divided into two subsections; the data pre-processing and the current forecast at HMS Networks. The first part is based on an analysis of the data received from HMS Networks and the steps of pre-processing that were necessary. These steps were conducted in the initial Design & Development step in cycle 1. The resulting data created the foundation for the training and testing of the ML-models in all three cycles. The current forecast at HMS Networks is described based on interviews and the SIOP presentation to gain an understanding of the current forecasting process which facilitated the comparison to the ML-models.

4.1 Data

The data provided by HMS Networks consists of sales data between 2021 and 2025, along with matching product data. Each observation in the sales data corresponds to the sale of a specific SKU, and the dataset comprises approximately 400 thousand observations. For every observation, there are 23 explanatory column values, describing important dates, shipping information and pricing associated with the transaction.

All SKUs are grouped into product groups, which is further categorized into product categories corresponding to the product branding. This hierarchical structure is reflected in the product data provided by HMS Networks, and illustrated in Figure 19. The product data also contains information about each SKU's life cycle state, the forecasting method currently in use and whether the product is tangible or not.

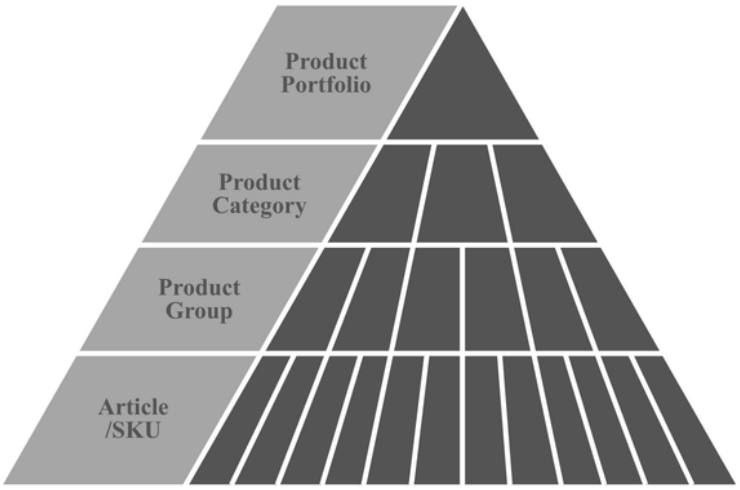


Figure 19: Illustration of the product taxonomy.

4.1.1 Filtering

In order to keep the thesis within the delimitations presented in Section 1.6, the data has been filtered to only include sales observations where the SKU fulfills certain criteria. These criteria were put together in collaboration with HMS Networks, to focus on the products where demand forecasting is necessary and with sufficient historical data. The criteria that were used are presented in Table 5. The filtering was done by matching the SKUs in the sales data with the corresponding product characteristics from the product data.

SKU Characteristic	**Accepted Value**
Life Cycle State	Active, Mature
Forecasting Type	Standard Sales Article, New Sales Article
Product Type	Physical, Physical with battery

Table 5: Filtering criterion used on the sales data.

4.1.2 Time Clumping and Zero-Fill

Since the main objective of the thesis was forecasting, the sales data was transformed into a time series format. This was achieved by accumulating the demand quantities for every unique SKU on a weekly or monthly basis. This method is illustrated in Figure 20. The weeks or months were counted continuously from the first date observed in the sales data.

In order to maintain a continuous time series, zero-fill was used to populate missing rows between observed demand. This method ensured a consistent temporal format of each series, but it assumes that missing values correspond to zero demand. This assumption may not always hold, as discussed in 3.1.2. Consequently, zero-filling may alter the underlying pattern within the demand data.

For product groups or categories introduced at a later stage, no zeros were assigned prior to the first observed sale, as these products represent the absence of product rather than absence of demand.

SKU	Date (week)	Quantity
AA	1/1-2021 (week 1)	2
BB	1/1-2021 (week 1)	4
AA	3/1-2021 (week 1)	1
BB	16/1-2021 (week 3)	8
CC	17/1-2021 (week 3)	7

SKU	Week Continuous	Quantity
AA	Week 1	2 + 1
AA	Week 2	0
AA	Week 3	0
BB	Week 1	4
BB	Week 2	0
BB	Week 3	8
CC	Week 3	7

Figure 20: Converting sales rows into a time series tabular format.

4.1.3 Grouping

The final step of the data processing was to accumulate the time series data by product group or product category. This step was also done in collaboration with HMS Networks, and was used to ensure sufficient historical data. Each individual SKU was associated with a the corresponding product group and product category from the product data, which allowed the demand quantities to be aggregated for all SKUs within the same group/category and week/month.

4.1.4 Data Splitting

In order to test each model on unseen data, the final year was preserved for testing. All data from previous years was used for training. Within the training data, an expanding window was used with five sequential splits and three months as the validation horizon. This approach preserves the temporal ordering of the data, while allowing the models and hyperparameters to be evaluated on multiple periods.

4.1.5 Hyperparameter Tuning

The hyperparameter tuning was performed using the tool Optuna. For each model, a limited search space was defined and 40 combinations of hyperparameters were tested. As discussed in Section 3.2.7, finding the globally optimal configuration is difficult due to the number of combinations. This approach provided a practical balance between computational cost and comparable model evaluation.

4.1.6 Stock-Outs and Promotions

In Section 3.1.2 the concepts of stock-outs and promotions, and how they affect forecasting, were discussed. Stock-outs cause the sales data to inaccurately reflect demand and promotions may increase the level of volatility in the sales data. Since neither of these are identifiable by solely analyzing the data, they may result in inaccurate forecasts.

HMS Networks does not experience frequent stock-outs affecting their ability to fulfill demand. Furthermore, lost sales due to buyers switching to a competitor is unusual due to the long-term contracts and unique products offered by HMS Networks (Interview with Market Manager, March 9th, 2026). The strategy of having long-term customers and focusing on quality and trust reduces the need for promotions. Other than individual deals with certain buyers, the concept of promotions does not appear at HMS Networks. Therefore, neither stock-outs or promotions were taken into account when pre-processing the data.

4.1.7 Resulting Data

The final processed data used for the thesis was in a tabular time series format, with size dependent on the time structure and aggregations chosen, illustrated in Table 6. This processing pipeline was designed to transform the raw sales and product data into a structured format for time series forecasting, while ensuring sufficient data consistency.

		Time splits	
		Weeks	**Months**
Grouped by	**Product Group**	10 384	2 394
	Product Category	5 487	1 265

Table 6: Number of rows in the final dataset depending on the configuration used.

4.2 HMS Networks' Current Forecasting Methodology

According to the internal company presentation, SIOP is a set of assumptions which enables the coordination of decision making between supply chain and divisions. All decisions are based on a 12 month forecast to establish a consensus demand plan. These forecasts are created once a month at HMS Networks.

The current methodology used to perform forecasting within HMS Networks varies depending on the product and market characteristics. However, for the subset of products considered in this thesis, the forecasting methodology is much the same and illustrated in Figure 21. When HMS Networks are to perform a

forecast, they use a top-down approach, estimating the demand at product category level. This is done by the responsible division for each category. This estimate must then be converted from category level down to SKU level, which is done by using a 6 month rolling mean distribution of the SKUs within the product category. The rolling window distribution excludes extreme observations, defined as demand values that exceed three standard deviations using the last three months of historical sales. This is done to prevent temporary spikes, such as those caused by one-off sales projects, from disproportionately influencing the allocations across SKUs.

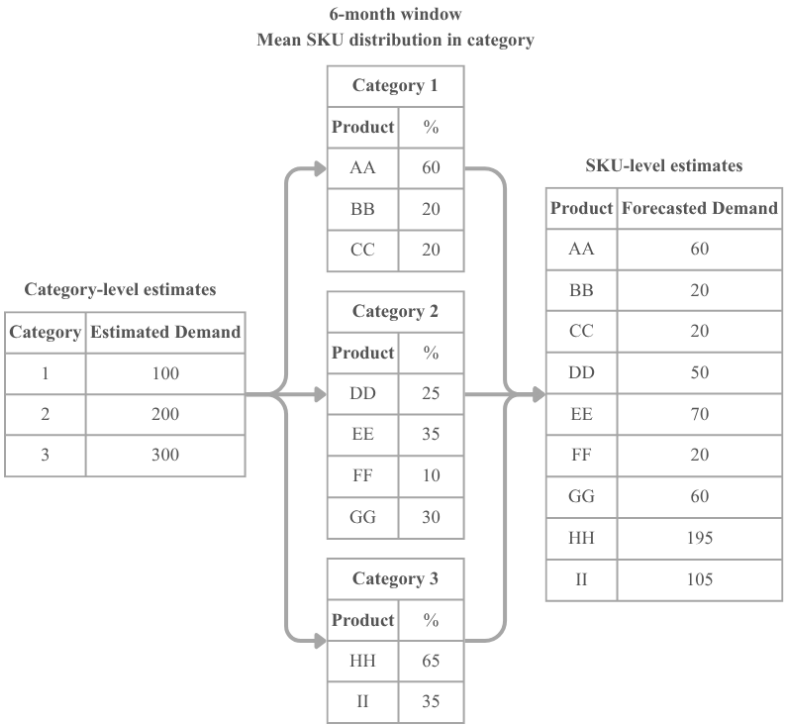


Figure 21: Disaggregation from category- to SKU level in HMS Networks' current forecasting methodology.

These forecasts are done at the start of every month, and cover a 12 month horizon. During the month, the forecast is reviewed and adjusted by the divisions, strategic planning and management. Adjustments made by strategic planning can be caused by different factors, for example:

- Customer orders exceeding the forecast
- Product sales patterns not recognized in forecast
- Unusually high or low sales during certain months affecting the forecast

5 Results

The results consist of four parts; the data, the benchmarks, the artifacts and the current forecast at HMS Networks. Firstly, the structure of the data following the pre-processing, described in the Empirics chapter, is presented. Secondly, the results of the benchmarks are calculated based on the aforementioned metrics. Thirdly, the results of the ML-models are presented. As discussed in Section 2.2.2, the development of the artifacts was done in three iterations, called cycles. The results of the artifacts are therefore presented per cycle, and then compared in a cycle summary. Thereafter, results of the ML-models based on data with different time and product level granularities are presented. Furthermore, the results of the ML-models are converted into SKU level forecasts and the corresponding metrics are presented. Finally, the performance of the current forecast at HMS Networks is calculated. The structure of the Results chapter is illustrated in Figure 22.

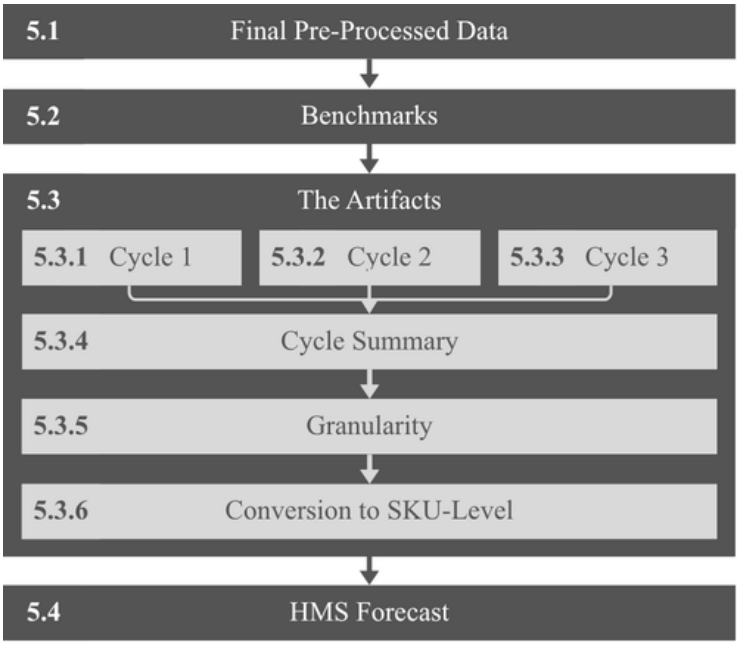


Figure 22: Illustration of the results chapter.

5.1 Final Pre-Processed Data

Within the data there was a total of 61 unique product groups and 30 unique product categories, all introduced at different points in time, as illustrated in Figure 23. In total, there were 20 months where all product groups were present and 6 months where all categories were present, or 20 months excluding the

product category introduced at month 56. Out of these 20 *complete* months, 12 were reserved for testing. As a result, the product groups and categories introduced latest only retained 8 months of training data. Since lagged features require historical observations prior to the prediction target, the maximum number of lagged features was 7, without introducing zero-fill and while retaining at least one training data point. Furthermore, the introduction of first-order differencing removed an additional data point from each series.

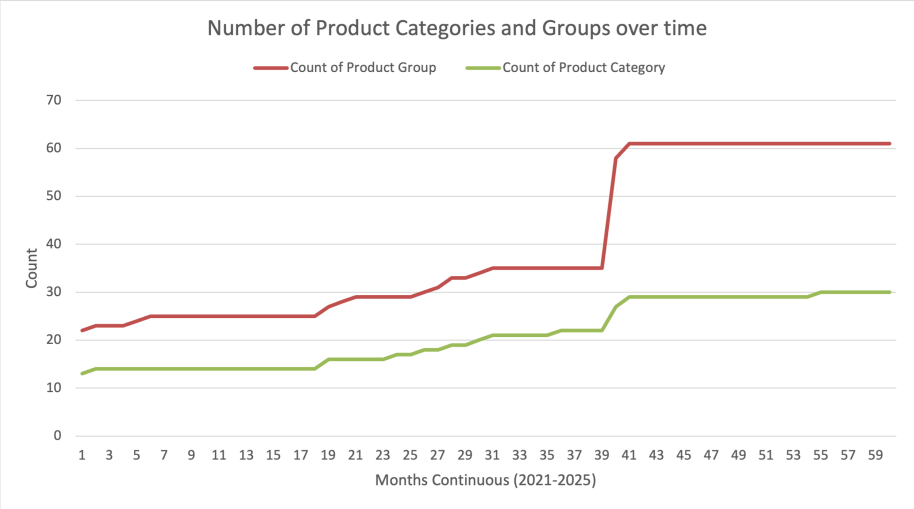


Figure 23: Product groups in the data over time, counted from the first occurrence in the sales data.

5.1.1 Amount of SKUs

The ML-models are, as previously mentioned, trained on sales data including approximately 2000 SKUs after filtering. The data of the current forecast at HMS Networks includes fewer SKUs than the sales data. To enable comparison between the artifacts and the current forecast at HMS Networks, the data must be filtered to only include the SKUs present in both data sets. This results in a data set which includes approximately 400 SKUs. Thus, when the results of the artifacts are presented it is based on approximately 2000 SKUs, while when the artifacts are compared to the current forecast the results are based on the 400 shared SKUs.

5.2 Benchmarks

The results of the benchmarks are shown in Figure 24. All benchmarks are calculated for the last 12 months of data, to be comparable to the forecasts. The corresponding metrics of every model are presented in Tables 7 and 8. The graph in Figure 24 shows the aggregated sales and benchmark values for all

products. The metrics were calculated for each benchmark, both on product group and product category level. Table 7 shows the metrics on a product group level and Table 8 shows the metrics on a product category level.

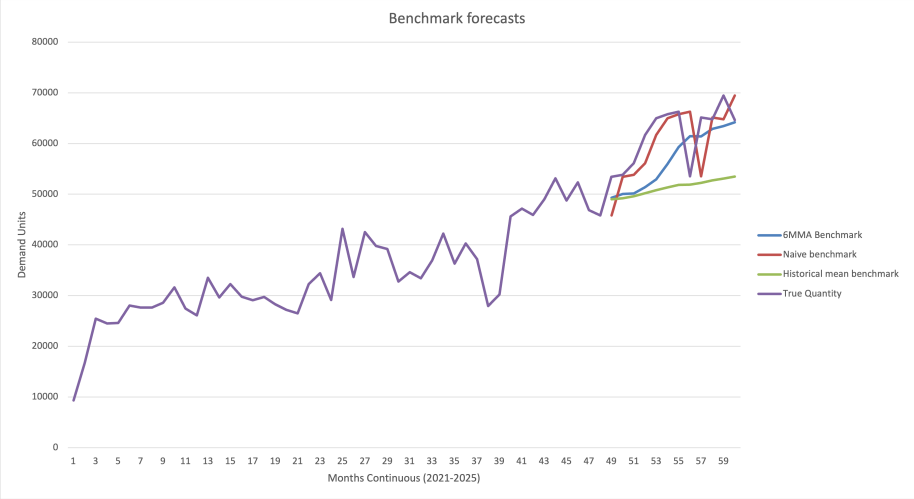


Figure 24: Benchmarks for a 12-month forecast.

Benchmark	**MAE**	**MAPE**	**RMSE**	**Bias**	**MAE + |bias|**
6M MA	236,3	109%	312	-78,1	321
Naive	280,7	90%	382,5	-25,7	313,3
Historical mean	310,5	109%	399,4	-169,8	572,7

Table 7: Metrics for the benchmarks on Product Group level.

The metrics in Table 7 show that all benchmarks have a negative bias, indicating an expected under-forecast by the benchmarks. Throughout all metrics the historical mean benchmark has the weakest performance. 6M MA and the naive benchmark have similar results for the MAE + |bias| metric, with naive performing slightly better. For the remaining four metrics, 6M MA and the naive benchmark show the best performance in two metrics each.

Benchmark	MAE	MAPE	RMSE	Bias	MAE + |bias|
6M MA	386,2	38%	472,0	-161,5	544,3
Naive	439,0	48%	558,5	-53,2	486,1
Historical mean	721,6	41%	808,4	-356,4	1374,9

Table 8: Metrics for the benchmarks on Product Category level.

The results in Table 8 are similar to the results in Table 7, with historical mean displaying the weakest performance throughout the majority of the metrics. The naive benchmark and 6M MA alternate at performing the best, with naive having the lowest score in the MAE + |bias| metric. The largest difference between product group and product category level is the MAPE metric, where the naive forecast has the strongest, respectively, weakest performance. As mentioned in Section 3.1.5, these benchmarks are calculated by only forecasting one month forward at a time. This allows for analysis on how a forecast performs on longer horizons, but deems the benchmarks inadequate as forecasts on their own.

5.3 The Artifacts

The initial results of the ML-models were done on a monthly basis and a product group level aggregation. The four models that were developed are; Decision Tree, Random Forest, LightGBM and XGBoost. In the upcoming sections the results from the three cycles are presented, showing the iterative development of the artifacts. As aforementioned, the three cycles focus on the structure of the data. Firstly, the models are trained on raw data (pre-processed as described in Section 4.1). In cycle 2, the data is further processed through FOD. Lastly, in cycle 3, the data is scaled. The process of FOD and scaling follows the descriptions in Section 3.3.3. All cycles were based on the recursive forecasting methodology presented in 3.3.4, and the global model structure presented in 3.3.5.

Following the three cycles, results of varying levels of granularity are presented for each of the four ML-models. Products are aggregated to both group and category level, and the time series is divided on both a monthly and weekly basis.

5.3.1 Cycle 1: ML-Models Without Feature Engineering

The forecast results from the trained ML-models are shown in Figure 25. Like the previous results, these predictions are aggregated over all product groups for every month. The hyperparameters found by Optuna, along with the corresponding validation score is shown in Table 9. In Table 10, the corresponding metrics for the machine learning models are illustrated. The metrics represent the mean over all product groups per month.

Decision Tree		***Random Forest***	
max_depth	16	max_depth	25
min_samples_leaf	6	max_features	0,393
min_samples_split	2	min_samples_leaf	9
		min_samples_split	8
		n_estimators	450
Validation MAE	**389,7**	Validation MAE	**324,2**

LightGBM		***XGBoost***	
colsample_bytree	0,656	colsample_bytree	0,642
learning_rate	0,016	gamma	4,015
min_child_samples	21	learning_rate	0,005
n_estimators	1400	max_depth	5
num_leaves	119	min_child_weight	1
reg_alpha	0,038	n_estimators	2500
reg_lambda	0,018	reg_lambda	0,001
subsample	0,845	subsample	0,739
Validation MAE	**390,0**	Validation MAE	**392,1**

Table 9: Hyperparameters and validation score of the ML-models in cycle 1.

From the results shown in Table 9, it can be seen that Random Forest has the lowest validation MAE, while the remaining models have similar scores. Figure 25 shows a tendency of all models to under-forecast compared to the actual demand. This is reflected in the metrics in Table 10, where all ML-models have large negative bias-values. Both Figure 25 and Table 10 show that the Random Forest model generates a forecast closest to the actual demand. Having the lowest score in all but one metric, Random Forest achieves the best performance of the four models during cycle 1.

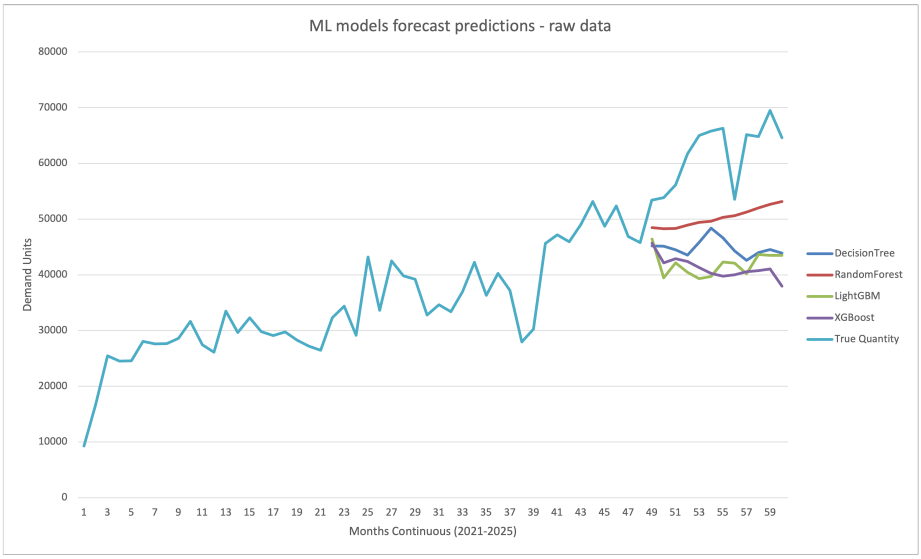


Figure 25: Aggregated forecasts from ML-models in cycle 1.

Model	MAE	MAPE	RMSE	Bias	MAE + |bias|
Decision Tree	456,8	136,1%	1676,7	-275,1	832,9
Random Forest	340,4	368,5%	1078,3	-186,7	619,8
LightGBM	457,0	455,4%	1643,1	-323,7	855,4
XGBoost	412,4	101,5%	1618,7	-334,8	775,7

Table 10: Testing metrics for the machine learning models in cycle 1.

5.3.2 Cycle 2: FOD

In the second cycle of the development, first-order differencing was used with absolute differences, as described in 3.3.3. The resulting hyperparameters and the corresponding validation scores are shown in Table 11. Among all models, XGBoost had the lowest validation score.

Compared to cycle 1, Figure 26 shows that the four ML-models perform more similar in cycle 2. Both Figure 26 and the corresponding metrics, shown in Table 12, demonstrate a reduced negative bias for all ML-models in cycle 2. Once again, Random Forest is shown to have the lowest MAE and MAE + |bias| score in the testing results from Table 12.

Decision Tree		***Random Forest***	
max_depth	2	max_depth	22
min_samples_leaf	5	max_features	0,780
min_samples_split	11	min_samples_leaf	3
		min_samples_split	2
		n_estimators	450
Validation MAE	**335,4**	Validation MAE	**319,5**

LightGBM		***XGBoost***	
colsample_bytree	0,752	colsample_bytree	0,804
learning_rate	0,001	gamma	3,271
min_child_samples	6	learning_rate	0,007
n_estimators	3800	max_depth	5
num_leaves	103	min_child_weight	2
reg_alpha	0,813	n_estimators	1900
reg_lambda	14,998	reg_lambda	0,399
subsample	0,768	subsample	0,766
Validation MAE	**326,3**	Validation MAE	**315,5**

Table 11: Hyperparameters and validation score of the ML-models in cycle 2.

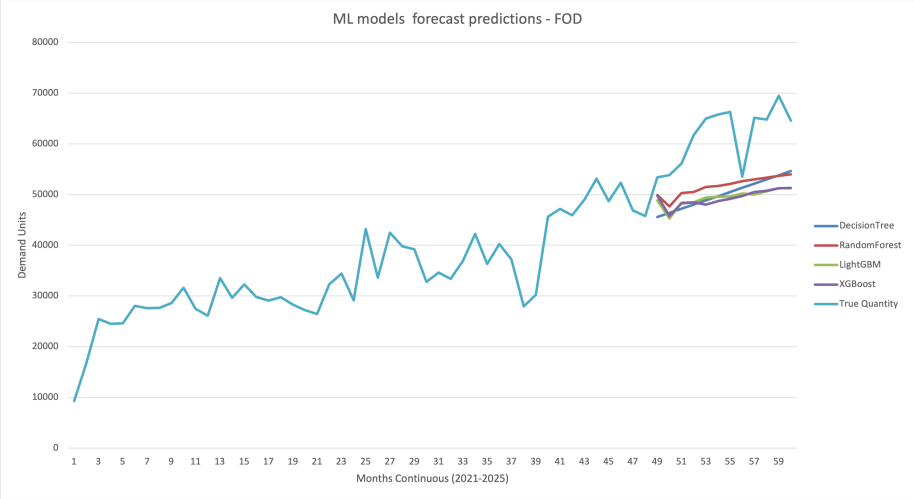


Figure 26: Aggregated forecasting estimates from ML-models using FOD.

Model	MAE	MAPE	RMSE	Bias	MAE + |bias|
Decision Tree	363,3	574,1%	1001,5	-189,3	676,9
Random Forest	301,9	362,3%	946,7	-163,0	544,9
LightGBM	310,9	308,6%	972,0	-200,4	574,0
XGBoost	328,7	408,9%	990,8	-202,2	609,8

Table 12: Testing metrics for the machine learning models in cycle 2.

5.3.3 Cycle 3: FOD and Scaling

In the final cycle, FOD was used in conjunction with scaling, as introduced in 3.3.3. The validation scores, shown in Table 13, show similar results for Decision Tree, Random Forest and XGBoost. From the aggregated forecasts in Figure 27, it is clear that Decision Tree overestimates the demand, while the other models appear to be closer to the true demand. This is further observed in Table 14, where Decision Tree has the highest MAE and MAE + |bias| metrics. The other three models have similar MAE and MAE + |bias| metrics, with a significant improvement from cycle 2.

Decision Tree		***Random Forest***	
max_depth	10	max_depth	18
min_samples_leaf	10	max_features	0,734
min_samples_split	19	min_samples_leaf	1
		min_samples_split	17
		n_estimators	900
Validation MAE	**345,9**	Validation MAE	**341,5**

LightGBM		***XGBoost***	
colsample_bytree	0,759	colsample_bytree	0,998
learning_rate	0,001	gamma	2,183
min_child_samples	5	learning_rate	0,003
n_estimators	3800	max_depth	8
num_leaves	108	min_child_weight	3
reg_alpha	0,850	n_estimators	2000
reg_lambda	18,747	reg_lambda	1,517
subsample	0,773	subsample	0,863
Validation MAE	**382,6**	Validation MAE	**343,9**

Table 13: Hyperparameters and validation score of the ML-models in cycle 3.

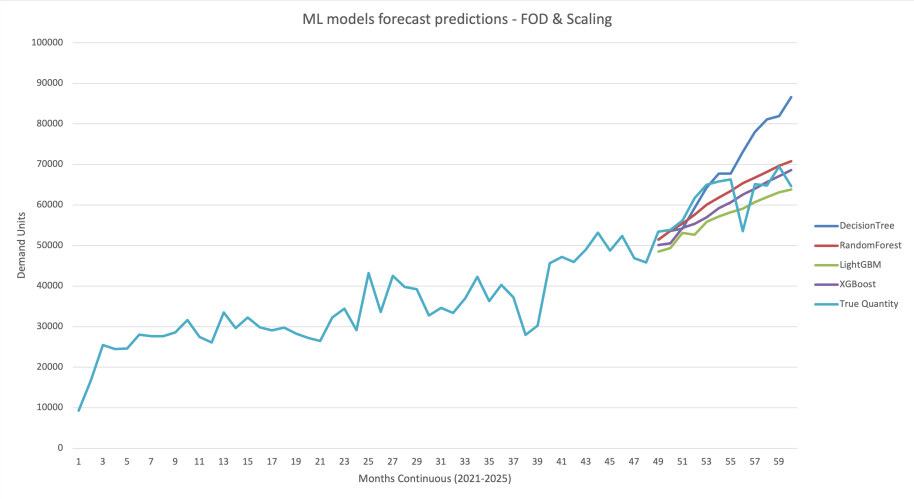


Figure 27: Reconstructed forecasting estimates from using first-order differencing and scaling.

Model	MAE	MAPE	RMSE	Bias	MAE + |bias|
Decision Tree	322,4	118,5%	771,1	108,1	521,4
Random Forest	261,2	113,9%	646,2	5,9	427,6
LightGBM	261,4	97,9%	696,3	-76,9	445,5
XGBoost	267,1	125,0%	702,4	-33,7	447,3

Table 14: Testing metrics for the reconstructed first-order differencing and scaled machine learning models.

5.3.4 Cycle Summary

Table 15 shows a summary of the validation results for each of the three cycles. The values shown are the MAE scores of the four ML-models. The lowest scores are found in cycle 2, while cycle 3 generally outperforms cycle 1 with Random Forest being the exception.

	Cycle 1	Cycle 2	Cycle 3
Decision Tree	389,7	335,4	345,9
Random Forest	324,2	319,5	341,5
LightGBM	390,0	326,3	382,6
XGBoost	392,1	315,5	343,9

Table 15: Summarized validation MAE by model and cycle.

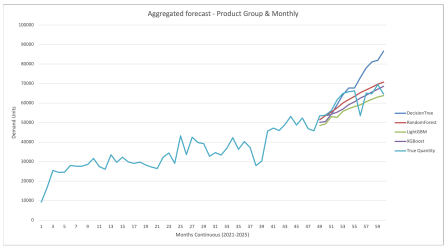
Table 16 summarizes the test scores for each of the three cycles. It shows the MAE + |bias| metric of the four ML-models. The first observation is that all four models generate improved test results after each subsequent cycle. LightGBM, which had the weakest performance after cycle 1, shows a significant improvement in cycles 2 and 3, becoming the model with the second best performance. XGBoost has a relatively low improvement between cycles 1 and 2, while improving significantly in cycle 3, achieving similar test scores to LightGBM. Decision Tree, while improving, achieves weak results in all cycles compared to the other three models. Lastly, Random Forest shows the best performance throughout all cycles.

	Cycle 1	Cycle 2	Cycle 3
Decision Tree	832,9	676,9	521,4
Random Forest	619,8	544,9	427,6
LightGBM	855,4	574,0	445,5
XGBoost	775,7	609,8	447,3

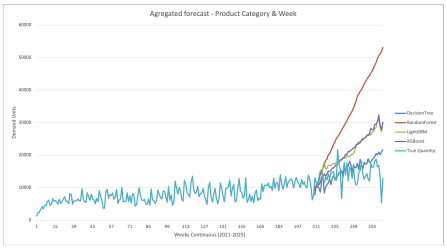
Table 16: Summarized test MAE + |bias| by model and cycle.

5.3.5 Granularity

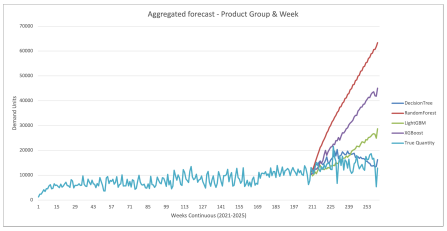
As discussed in Section 3.1.2, the granularity of time and product level has a large impact on a demand forecast, both in performance and usefulness in a SIOP process. To evaluate the impact of granularity has on the four ML-models, four different setups of data aggregation were created. In Figure 28a, the product data was aggregated to product group level on a monthly basis. In Figure 28b, the product data was aggregated to product category level on a weekly basis. In Figure 28c, the data was aggregated to product group level on a weekly basis. Lastly, in Figure 28d, the product data was aggregated to a product category level on a monthly basis.



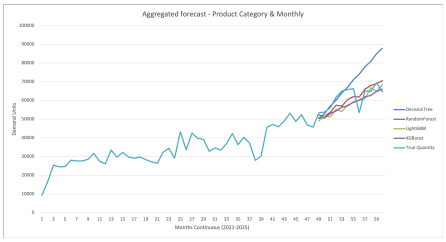
(a) Product Group - Monthly



(b) Product Category - Weekly



(c) Product Group - Weekly



(d) Product Category - Monthly

Figure 28: Comparison of the four model plots.

The graphs in Figure 28 show that the different levels of granularity have a significant impact on the performance of the ML-models. Figures 28a and 28d show that the models perform well when aggregating the data to months, with the exception of Decision Tree, which over-forecasts during the last six months. Figures 28b and 28c show the opposite result when aggregating the data to weeks, where the other three models over-forecast substantially more than Decision Tree. A further observation is that the models deviate significantly from the actual demand when forecasting weekly compared to monthly.

The MAE + |bias| metric of the two scenarios with monthly forecasts in Figures 28a and 28d is seen in Table 17. To be comparable, the metric has been recalculated for the product group scenario by aggregating the results into their corresponding category. The original granularity metrics can be seen in Appendix B. This results in the product category scenario achieving slightly better test scores for all ML-models except Random Forest, which achieves better scores on a product group aggregation. The best results are achieved by LightGBM in the product category scenario.

	Product Group	**Product Category**
Decision Tree	959,4	878,6
Random Forest	805,3	876,1
LightGBM	865,2	745,2
XGBoost	862,8	789,2

Table 17: MAE + |bias| of product group and product category aggregation on month granularity.

5.3.6 Conversion to SKU Level

For HMS Networks to be able to apply the forecast in their SIOP process, it must have the highest level of product level granularity. As discussed earlier, the forecasts have aggregated the products to product group and product category levels. A necessary step is, therefore, to convert these forecasts to SKU level. The conversion is done with the same method as for HMS Networks, which is by calculating the 6-month rolling shares of all SKUs within each product group/category, as mentioned in Section 4.2. The error metrics were then calculated for each ML-model. Table 18 shows the MAE, bias and MAE + |bias| metrics for the monthly forecasts of both the product group and the product category aggregations for all four ML-models.

	Product Group			Product Category		
	MAE	Bias	MAE + |bias|	MAE	Bias	MAE + |bias|
Decision Tree	24,2	4,9	40,6	24,2	5,4	40,6
Random Forest	21,8	1,3	35,8	21,8	0,6	35,9
LightGBM	20,9	-1,7	33,9	20,9	-0,9	33,7
XGBoost	21,2	-0,1	34,6	20,8	-1,0	33,7

Table 18: Error metrics of product group and category converted to SKU.

When comparing the product group forecast and the product category forecast, the results are practically identical. Decision Tree shows the weakest performance due to a general tendency to over-forecast. The other three models have a bias near 0 for both product aggregation cases, with Random Forest slightly over-forecasting and the boosting models slightly under-forecasting. These three models have similar MAE as well as MAE + |bias| scores in both cases. Due to volatility, the scores are too similar to identify a significantly superior model. However, based on the results in Table 18, the models that perform the best are LightGBM and XGBoost when forecasting based on product category and month.

5.4 HMS Forecast

As a final benchmark to determine the performance of the ML-models, the results of the current forecast at HMS Networks were analyzed. In Figure 29 the 12-month forecast is compared to the actual sales data of 2025. Similar to previous figures, the quantities of all product categories are aggregated.

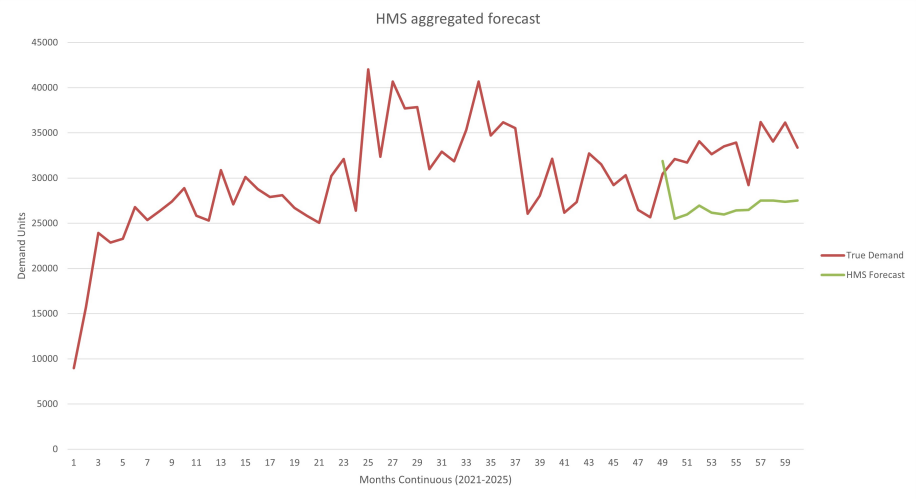


Figure 29: HMS forecast aggregated by month and product category.

To enable a quantitative comparison to the ML-models, the error metrics of the HMS forecast were calculated on a product category and SKU level. The results of the calculated metrics are seen in Table 19. The significantly negative value of the bias metric indicates that HMS Networks tends to under-forecast considerably. This is confirmed by Figure 29, which shows that the HMS forecast is lower than the actual sales data throughout the last 11 months of the year. The resulting SKU level errors metrics, only including SKUs found in both forecasts, are found in Appendix C.

Aggregation level	MAE	Bias	MAE + |bias|
Product category	585,4	-400,5	1043,4
SKU	45,6	-14,8	77,2

Table 19: Error metrics for the HMS forecast calculated on category and SKU level.

6 Discussion

The results presented in Chapter 5 provide a quantitative evaluation of the ML-models. However, numerical results alone do not provide sufficient insight into why certain models perform better than others. Hence, this chapter aims to interpret and discuss the numerical results and contextualize them by linking them to the frame of reference and the methodology choices made throughout the study.

The disposition of the discussion is visualized in Figure 30. Firstly, the characteristics of the final pre-processed data are interpreted. Thereafter, each development cycle of the artifacts is analyzed, followed by a discussion of the impact of granularity and aggregation on the results. Lastly, a comparison of the artifacts and the current demand forecast at HMS Networks is carried out.

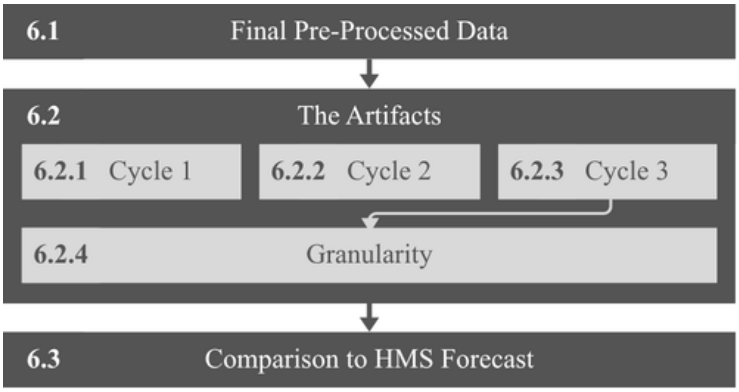


Figure 30: Illustration of the discussion chapter.

6.1 Final Pre-Processed Data

It is essential to understand the characteristics of the final pre-processed dataset in order to interpret and understand the performance of the developed ML-models. As highlighted in Section 3.1.2, the structure and quality of demand data have a significant impact on forecast performance, particularly when characterized by high variability and intermittent demand.

Through the pre-processing steps, the raw data was transformed into a suitable format for ML. The filtering reduced the dataset to only contain SKUs deemed appropriate and important for the study, in accordance with the frame of reference and input from HMS Networks. This primarily resulted in a focus on standard products, which generally exhibit a more stable demand pattern compared to customer specific or new products. In addition to this, HMS Networks considered these SKUs to be the most relevant for forecasting and planning activities.

Although the filtering step improved the overall relevance and stability of the dataset, several challenges related to data quality and representation remain. The implementation of zero-filling for periods without any sales, while necessary to create consistent time series, altered the original data by assuming that periods without recorded sales correspond to zero demand. In reality, such periods may also reflect stock-outs or other constraints, meaning that the transformation risked introducing a systematic underestimation of true demand. However, as mentioned above, stock-outs are an uncommon occurrence at HMS Networks, deeming zero-filling to be a reasonable method of processing the data.

The relatively low number of training months in the pre-processed data was also noted as a limiting factor for model performance. In order to retain the 12-month forecast horizon, the product category introduced at month 56 had to be excluded, since it otherwise would not provide sufficient historical observations for both training and testing. Even with this exclusion, the latest introduced product groups and categories only retained a total of 8 months of training data. This imposed a limitation on the maximal number of lagged features, as the lagged observations and remaining data points could not exceed this.

In addition to data representation, the level of granularity significantly affects the forecasting performance. While higher levels of granularity reduce variability and improve stability, more granular data introduces noise. On the contrary, a more granular forecast is closer to the level at which the decisions are made, and therefore requires less disaggregation. This trade-off is particularly relevant in the context of this study, as the forecasts are ultimately required at SKU level, and is therefore further discussed in Section 6.2.4.

6.2 The Artifacts

The developed ML-models represent the artifact of this study and were constructed through an iterative DSR process as described in Section 2.2.2. Each cycle introduced modifications to the data representation, with the aim of improving the performance of the forecasts. The following sections analyze each cycle individually, focusing on how the applied changes influenced forecasting performance and what insights can be drawn regarding the application of ML-models for demand forecasting.

The results of each cycle are compared to the previous cycle to identify any improvements. Furthermore, the results are compared to the three benchmarks; naive benchmark, historical mean and 6M MA, for a consistent baseline comparison. As mentioned in Section 3.1.5, the benchmarks are separately calculated for each month of the 12-month forecasting window, based on the most recent sales data. Compared to a 12-month forecast, which is expected to decrease in accuracy for later months, the benchmarks have the same conditions for each month. Therefore, the benchmarks are inapplicable as forecasts in practice, but offer insights into how well a forecast is expected to perform based on data volatility and trends.

6.2.1 Cycle 1

The first cycle represents the initial implementation of ML-models on the data without additional feature engineering. The models were trained on the pre-processed dataset in its default form, providing a baseline for evaluating the subsequent cycles.

When comparing the different models, Random Forest had the best performance on both the validation and test metrics, with MAPE being the only exception. When comparing validation and test scores, the MAE metric was relatively unchanged for all ML-models. This indicates that the models did not overfit to the training data, thus suggesting that the models had an ability to generalize trends and patterns in the data. However, despite the ability to generalize, the models had a weak performance compared to the benchmarks.

The best performing ML-model during cycle 1, Random Forest, had a larger bias, MAE and RMSE than the worst performing benchmark, historical mean. All ML-models showed a tendency to significantly under-forecast, causing the large negative bias. The high values of MAE and RMSE indicate a large number of significant numerical errors. However, XGBoost performed better than 6M MA and historical mean according to the MAPE metric. This could suggest that XGBoost was better than the other three ML-models at forecasting product groups with low sales quantities.

Despite Random Forest achieving the best overall performance, one limitation can be observed across all models. From Figure 25 (p. 67), along with the consistent under-forecasts, it seems that none of the models were able to capture the trend. This may be attributed to the structure of tree-based models, where predictions are generated as averages within terminal nodes. As a result, the models produced piecewise constant forecasts, which limited their ability to follow trends outside of the observed range. The ML-models, therefore, avoided forecasting higher values than previously observed values in the training data, causing the models to plateau rather than identify the positive trend of the sales data.

This indicates that the limitations in cycle 1 are not exclusively a consequence of the data characteristics, but also of how the models were structured. While the ensemble nature of Random Forest enabled it to reduce variance in the estimates, it did not resolve the fundamental issue of trend representation. This limitation is particularly relevant in the context of demand forecasting, where trends are common segments of a products life-cycle. Without the ability to capture such dynamics, the models were restricted to reproducing historical demand levels rather than adopting to changing demand patterns.

Furthermore, the hyperparameter tuning process provided additional insights into model behavior. As mentioned in the frame of reference, hyperparameters are linked to the complexity of the model. More complex configurations increase the risk of overfitting to the noise in the data, which would appear as large dis-

crepancies between validation and test performance. On the contrary, a model must be complex enough to capture the underlying patterns. However, the hyperparameter values themselves are not sufficient to draw conclusions regarding performance or behavior. Instead, their relevance lies in how they evolve across cycles in relation to changes in data representation and model performance. Hence, the hyperparameters in Table 9 (p. 66) were used as a baseline for which the following cycles hyperparameters will be compared against.

Overall, cycle 1 demonstrated that while Random Forest was able to handle variability better than other models, all models were limited by their inability to capture trends. These findings suggest that improving the forecast performance is not solely a matter of model selection, but also of transforming the data into a representation that aligns with the structure of the models.

6.2.2 Cycle 2

The second cycle aimed at resolving the fundamental limitation identified in cycle 1, that tree-based models were unable to capture trends. To address this limitation, first-order differencing was implemented as a data transformation, making the models estimate changes rather than demand. By removing trend components and stabilizing the mean of the series, the transformation aimed to make the data more suitable for the tree-based models.

The results from cycle 2 indicate a consistent improvement in model performance compared to the baseline established in cycle 1. Across all models, validation MAE was reduced, suggesting that the application of first-order differencing made the underlying patterns in the data more accessible for the models. In addition, the performance difference across models was decreased in cycle 2, with all models achieving relatively similar validation scores, with XGBoost achieving the best score. Random Forest followed closely with validation MAE 1.27% worse than XGBoost.

When analyzing the test results, a similar trend is observed. All models achieved improved performance according to most metrics compared to cycle 1. Despite an improvement from cycle 1, the bias score remained negative for all ML-models, as shown in the test scores and the aggregated graph in Figure 26 (p. 69). The differences between the models remained relatively small, and Random Forest achieved the best results across all metrics except MAPE, where LightGBM achieved the best score. Although, all MAPE scores were significantly worse than during cycle 1, with the best value increasing from 101,5% (being XGBoost in cycle 1) to 308,6% (being LightGBM in cycle 2).

The improved performance could be explained by the effect of first-order differencing on the data representation. By transforming the time series into changes rather than absolute levels, the models are no longer required to capture trend components directly. Instead, the models are able to focus on learning local variations in demand, which better matches the capabilities of tree-based models. In addition, this should allow models to predict demand higher than has been

observed in the historical data. This can be seen in the bias metrics from the test results, which have been reduced compared to cycle 1. However, the monthly changes predicted by the models are still limited to the changes observed in the training data, similar to the limitations of absolute level estimates in cycle 1. Furthermore, the high values of the MAPE metric is likely due to the models performing worse when forecasting product groups with low quantities.

Despite showing improved results in cycle 2, with the exception of MAPE, the ML-models still performed worse than the naive and 6M MA benchmarks. Furthermore, all models except Random Forest showed weaker test scores than the historical mean benchmark. Random Forest, however, scored better than the historical mean in the MAE, bias and MAE + |bias| metrics. These results further indicate an improvement from cycle 1, while simultaneously suggesting that further improvements of the ML-models should be pursued.

The hyperparameter tuning results gave further insight into the behavior of the models. For instance, the Decision Tree model in cycle 2 was found optimal using a maximum depth of 2, compared to 16 in cycle 1, while achieving better scores in general. This suggests that the transformation allows the model to understand patterns using a simpler model structure, which may reduce the risk of overfitting. While this is true for the Decision Tree, the other models remained fairly similar in their structure, which indicates that the relationships in the data may still be complex to capture.

Another observation is the relative linearity of the forecasts in cycle 2. This may be a result of optimizing the models with respect to MAE. As discussed in the frame of reference, MAE does not penalize the large deviations as strongly as metrics such as RMSE. Instead, MAE tends towards median predictions, which would result in more linear forecasts when using FOD.

While the findings of cycle 2 point toward an improvement compared to cycle 1, there are still notable limitations. The continued under-forecasting of demand and poor MAPE scores suggest that the models were unable to accurately represent the magnitude of the demand. One possible explanation for this is, as mentioned, the varying demand magnitudes. Some product groups exhibit much larger quantities than others. This could result in the models not being able to effectively learn patterns cross-series, and hence not being able to generalize well.

6.2.3 Cycle 3

The third cycle aimed to improve upon the limitations found from cycle 2, in which it was found that the varying demand magnitudes across the product groups could be negatively impacting the models' ability to find patterns cross-series. As mentioned in Section 3.3.3, scaling the data is essential for cross-series learning. To address this, cycle 3 combined first-order differencing with scaling. By scaling the differenced values relative to the mean series demand, the influence of magnitude could potentially be reduced, creating a more uniform

representation across the product groups. This may increase the models' ability to generalize cross-series, rather than being biased toward groups with larger magnitudes and hence larger impact on the overall score.

The outcome of cycle 3 is more nuanced than the previous cycles. While the overall validation scores are slightly worse than in the second cycle, most models still have better validation performance than in the first cycle, with the exception of Random Forest which achieves its lowest validation score in cycle 3. This suggests that the introduction of the scaling transformation does not improve all models' performances but rather affects the models differently depending on their structure and sensitivity to magnitude.

An interesting observation in the results is the relationship between validation and test MAE. In previous cycles, validation MAE has been lower than test MAE, but cycle 3 shows the opposite relationship for all models. This is somewhat counterintuitive as validation scores are generally expected to align with the test performance, and in particular not to underestimate it. One possible explanation for this behavior is that the validation sets are more challenging than the test set, for instance, due to higher variance. While scaling creates a consistent scale across product groups, it does not eliminate the underlying differences in the data distributions. Hence, some periods may still be more difficult for a given model to accurately forecast, and underestimate the models true ability to generalize. This suggests that the validation metric alone does not fully capture the models performance in this setting.

When comparing the models, Random Forest once again achieves the best overall test results. However, the MAE scores of LightGBM and XGBoost are very similar to that of Random Forest, which is also reflected in their MAE + |bias| metrics. This consistency across models suggests that the applied data transformations play a central role in improving performance, effectively reducing differences between model structures. As in cycle 2, this indicates that the gains are driven primarily by the improved data representation, rather than by the choice of model itself.

Another improvement was noted in the aggregated product group forecasts in Figure 27 (p. 71). The total forecast demand, in terms of magnitude, was much closer to the actual demand compared to previous cycles. This was further supported by the bias metrics in the test results being smaller than in previous cycles. This supports the hypothesis that differences in magnitudes across series were limiting the models' ability learn patterns within the data, rather than understanding magnitudes of the various product groups. Furthermore, the process of scaling the data resulted in a significant improvement of the MAPE metric. This is likely due to the models being able to weigh each product group equally, independent of the varying demand magnitudes.

The improvements made in cycle 3 were also reflected in the comparison to the benchmarks. The best performing model, Random Forest, scored a test MAE only 10.5% worse than the best performing benchmark, 6M MA. Additionally,

it reached a better bias score than all benchmarks. However, the other test metrics were, in general, slightly worse than the benchmarks, with RMSE being significantly higher.

The hyperparameters in Table 13 (p. 70) show a general increase in model complexity compared to cycle 2. For instance, the max depth of the Decision Tree rose to 10, while the Random Forest used double the number of estimators but with a lower max depth. LightGBM and XGBoost remained relatively stable in their configurations. This could suggest that, after normalization, the models were better able to utilize the more complex structures to capture relationships within the data. However, as in previous cycles, no definitive conclusion can be drawn regarding the impact of the hyperparameters on the produced forecasts.

Another important aspect of the scaling implementation is the choice of scaling factor. In this cycle, the series mean within the training data was used, but many other options exist. Examples of other potential scaling factors are series maximum, minimum or median. All of these alternatives would generate different datasets, thus creating forecasts with different results. However, the main objective remains the same for all alternatives, which is to make each series comparable despite varying magnitudes.

Overall, the results from cycle 3 suggested that combining first-order differencing with scaling improved the models' ability to identify and represent both temporal dynamics and demand magnitudes. Although the validation performance did not improve, the test results suggested better generalization and a reduced bias. This highlights the importance of scaling when working with multi-series data, as Section 3.3.3. Together with the previous cycles, these findings show that the improvements in forecasting performance are greatly affected by aligning the data representation with the models' abilities. This suggests that forecast performance is largely driven by the data representation, rather than model selection.

6.2.4 Granularity

The results from the granularity tests further highlighted the impact of the data representation on the forecasting performance. As noted from Figure 28 (p. 73), the level of granularity significantly affected both the stability and the accuracy of the forecasts. In general, higher levels of granularity introduce increased variability and forecast difficulty, while lower levels appear more stable.

A notable observation is the divergence of the weekly forecasts. These models diverged significantly from the true demand, leading to substantial overestimates. One possible explanation is the forecasting methodology used. The use of recursive forecasts could potentially increase the risk of divergence, as small errors accumulate over time. This effect is particularly notable for longer forecasting horizons, since small errors would be able to propagate further. Another explanation of the divergence could be the relatively small number of time lags

compared to the forecasting horizon. As mentioned in Section 6.1, each model could, at best, utilize 7 months of time lags (or 28 weeks), with at least one training data point, when attempting to predict a full year. Consequently, a large portion of the forecasting horizon was generated without direct reference to actual observations, possibly making the recursive predictions increasingly sensitive to accumulated errors.

Additionally, the first-order differencing and scaling may have further increased the sensitivity to small prediction errors. The trending components in the weekly granularity were less significant, possibly enabling simpler data representations to perform better. In other words, the benefits of FOD and scaling might be reduced with higher temporal granularity, and instead increase the risk of errors propagating.

Notably, the simplest model, the Decision Tree, actually performed significantly better than the complex models on the weekly data. This may indicate that the weekly data contains a higher level of noise and variation, which the simpler model was less likely to overfit to. Under such conditions, more complex models may be disadvantageous, as they are more likely to overfit to noise.

When analyzing the monthly forecasts, the results from the product group aggregation were converted to category level to facilitate the comparison to product category aggregation. The resulting test scores showed similar performance by the two levels of product granularity, with product category aggregation being slightly superior in general. When the estimates were converted to SKU level, using the methodology described in Section 4.2, the results were even more similar. The best results, according to the MAE + |bias| metric, were found from LightGBM and XGBoost with product category granularity.

The similarity between the results of both product group and product category aggregation indicates that product level granularity has less effect on the performance of the forecasts than temporal granularity. This suggests that the quantities are large enough in the product groups to enable a relatively accurate forecast despite larger variation. Worth noting is that there are other methods to disaggregate the results to SKU level. The same method as HMS Networks utilizes was applied to facilitate comparison between the forecasts, although other disaggregation methods could potentially generate better results.

Compared to the benchmarks, all ML-models generated on a monthly basis with product category granularity had better MAE + |bias| metrics than historical mean. However, none of the models performed better than the naive, nor the 6M MA benchmarks. Given that the benchmarks are rolling one-month forecasts, while the ML-models generate full 12 month forecasts, these results still indicate promising performance from the ML-models.

To summarize, granularity has a significant impact on the resulting forecasts. The increased temporal granularity generated divergent forecasts and is therefore not appropriate for the given forecasting context. The lower level of both temporal and product level granularity generated better results, likely due to

reduced noise and variance in the data. These findings further support the importance of granularity in ML-based demand forecasting.

6.3 Comparison to HMS Networks' Forecast

To further analyze the performance of the ML-models, they are compared to the current forecast at HMS Networks. The current forecast offers a representation of the difficulties of forecasting in the specific industry context of the company, allowing for a deepened understanding of the performance of the ML-models. As mentioned in Section 5.1.1, the number of overlapping SKUs between the sales data and the forecast by HMS Networks is significantly lower than the previously applied dataset. Therefore, the results of the ML-models were filtered and the metrics recalculated to allow for an accurate comparison. Thus, the filtered dataset consists of 407 SKUs, approximately 20,5% of the previously included SKUs.

The updated metrics of the ML-models are shown in Tables 20 and 21 together with the corresponding metrics of the HMS forecast. To be comparable to the forecast by HMS Networks, the forecasts are based on month and product category. Table 20 shows the MAE, bias and MAE + |bias| metrics calculated on the category level. The metrics for each individual category are presented in Appendix D.

	MAE	Bias	MAE + |bias|
DecisionTree	759,3	470,3	1352,4
RandomForest	603,2	193,1	1077,2
LightGBM	510,2	77,6	871,1
XGBooost	486,0	61,8	844,3
HMS	585,4	-400,5	1043,4

Table 20: Error metrics for ML-models and HMS forecast (on monthly and product category granularity).

Table 20 shows that both LightGBM and XGBoost performed better than the forecast by HMS Networks on all three metrics. XGBoost generated the best test scores when applied to the updated dataset. Random Forest had a smaller bias than the HMS forecast, while showing a poorer performance on the MAE and MAE + |bias| metrics. Decision Tree exhibited the weakest test scores, performing worse than the HMS forecast across all three metrics.

Notably, all ML-models generated a positive bias, compared to the HMS forecast which showed a significantly negative bias. This indicates a tendency of over-forecasting across all ML-models. The bias is visualized in Figure 31, where all product categories are aggregated. XGBoost and LightGBM, which generate the

best bias scores, under-forecast slightly during the first few months, while over-forecasting following the dip in sales occurring in month 56. The significantly lower bias compared to the HMS forecast indicates the ability of XGBoost and LightGBM to more accurately identify the baseline demand.

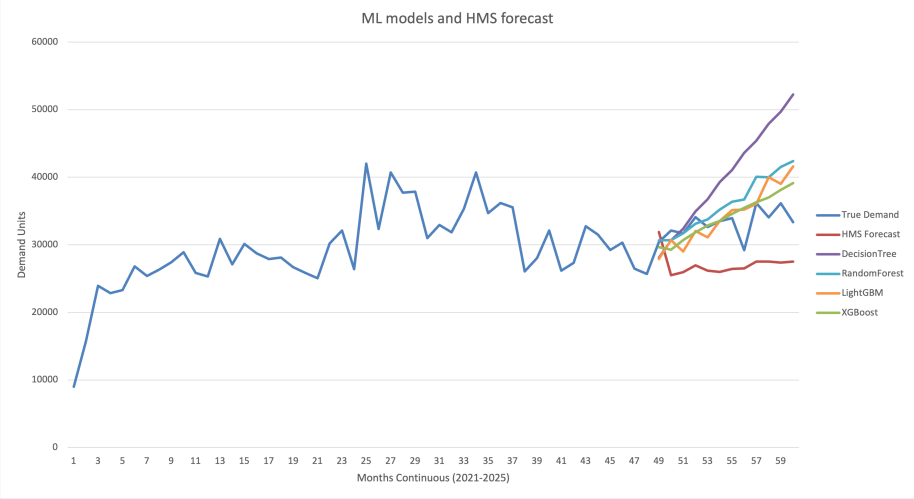


Figure 31: ML-models and HMS forecasts (month and product category aggregation) using filtered SKUs.

Table 21 shows the results of the forecasts after disaggregating to SKU level. The findings are similar to those of Table 20, with XGBoost performing the best, followed by LightGBM. Random Forest generated a lower bias than the HMS forecast, while performing worse in the MAE and MAE + |bias| metrics. Lastly, Decision Tree produced the weakest scores across all metrics.

Interestingly, the MAE + |bias| score of the best performing ML-model, XGBoost, was only 6% better than the HMS forecast. However, XGBoost still scores significantly better than the HMS forecast when it comes to the bias metric.

	MAE	Bias	MAE + |bias|
DecisionTree	53,5	17,3	90,7
RandomForest	47,6	7,1	79,2
LightGBM	45,0	2,9	73,4
XGBoost	44,4	2,3	72,4
HMS	45,6	-14,8	77,2

Table 21: Error metrics for ML-models and HMS forecast (on monthly and product category granularity).

To summarize, the HMS Networks forecast considerably under-estimated the true demand while best performing ML-models, LightGBM and XGBoost, managed to perform better. This was much due to a lower average bias. However, both Decision Tree and Random Forest performed worse according to the MAE + |bias| metric. These results were also reflected at SKU level, although with smaller differences to the HMS forecast. These results indicate that the use of ML-models to forecast demand has the potential to be beneficial in the context of HMS Networks.

7 Conclusion

The purpose of this study was to explore how ML-models could forecast demand within a complex product portfolio at HMS Networks. More precisely, the study aimed to evaluate if ML-models could improve forecasting performance compared to traditional methods, as well as to investigate the effects of data aggregation and granularity. Based on the findings from Chapter 5 and Chapter 6, this chapter summarizes the main conclusions and their consequences.

The disposition of the conclusion is visualized in Figure 32. Firstly, the three research questions are answered. Secondly, the contributions of the thesis are concluded. Thirdly, the limitations of the thesis are discussed followed by recommendations for future research. Lastly, the possibilities and challenges of a practical implementation at HMS Networks are raised.

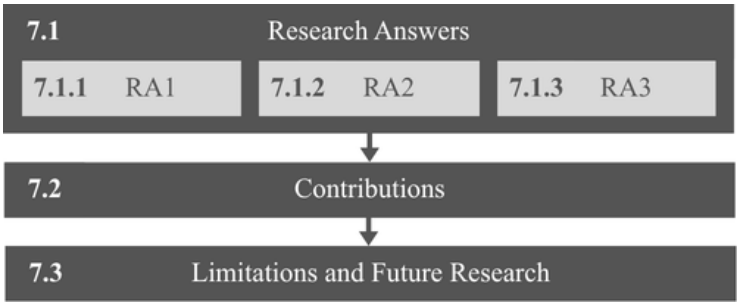


Figure 32: Illustration of the conclusion chapter.

7.1 Research Answers

The discussion in the previous sections has touched on several aspects of the results and the research process. In the upcoming sections, the research questions presented in Section 1.5 are discussed and answered based on the results in Chapter 5. As outlined above, the research questions were as follows:

1. *Which ML-models can be used to improve demand forecasting of a complex product portfolio?*
2. *What levels of granularity (temporal and product hierarchy) offer the best conditions to generate a forecast on SKU level?*
3. *How would a ML-based forecasting method perform compared to the current forecasting method used at HMS Networks?*

7.1.1 Research Answer 1

The first research question raises the question of which ML-model is most suitable to forecast demand in a complex context. The question is answered in two steps during the research process. Firstly, the previous literature on the subject of machine learning as a tool for demand forecasting offers a generalized answer. Secondly, the test results of the developed artifacts offer an answer for the context of HMS Networks within the delimitations of the thesis.

The review of literature, presented in Section 3.3.6, showed that ML-models based on Decision Trees yield the most promising results. Furthermore, no specific model could be determined as universally superior, rather the optimal model depended on the specific context of the demand that was forecast. However, the model which performed the best in the largest number of studies was XGBoost.

In the thesis, the ML-models; Decision Tree, Random Forest, LightGBM and XGBoost were developed. The models were then applied to the case of HMS Networks. The results showed that several models performed well and with similar scores of the MAE + |bias| metric. Random Forest, LightGBM and XGBoost achieved similar results on product category, product group and SKU level. Rather than highlighting a single ML-model as optimal, the results showed that the performance of each model could be improved by structuring the data to create optimal model training conditions.

When structuring the data to achieve the best metrics scores (applying FOD and scaling as well as aggregating to month and product category granularity), Random Forest performed the best, followed by LightGBM and XGBoost. However, the results showed that decisions regarding data structure strongly affected the performance of the ML-models. For example, when forecasting on a weekly basis, Decision Tree achieved the best results, despite performing the poorest in the monthly forecasts.

To answer the first research question, no single ML-model can be determined as superior to the others. The best results were achieved with Random Forest. However, Random Forest, LightGBM and XGBoost all achieved promising results when forecasting demand in the case of HMS Networks.

7.1.2 Research Answer 2

The second research question focuses on which levels of temporal and product level aggregation are suitable for the development of ML-models for demand forecasting. As mentioned in Section 6.2.4, the granularity of the time series data has a substantial impact on the performance of forecasting models. In this case the granularity of time intervals and product aggregation were of interest. The ML-models were tested at time intervals of weeks and months, and product aggregations of product group and product category.

The results showed that the monthly forecasts performed noticeably better than the weekly forecasts, due to a tendency of the ML-models to severely over-forecast when forecasting based on weeks. This was mainly attributed to the higher variability and noise of the weekly data, and the recursive forecasting methodology allowing the models to diverge.

Product level aggregation, however, did not affect the results to the same extent as temporal aggregation. Therefore, both a monthly product category based solution and a monthly product group based solution would be suitable in the case of HMS Networks. When disaggregating the forecasts to SKU level, the results of the two product aggregation scenarios generated nearly identical metric scores.

To answer the second research question, temporal aggregation had a significantly larger impact on the results than product level aggregation. Aggregating the data by month showed more promising results than by week. Thus, both aggregation to month and product group as well as aggregation to month and product category are determined as good conditions to generate a ML-based forecast.

7.1.3 Research Answer 3

The third research question aims to determine how an ML-model would perform in the context of HMS Networks in comparison to their current forecasting methodology. As mentioned in Section 1.5, the question is divided into two parts. Firstly, the results of the ML-models are compared to the HMS forecast. Secondly, the implications of a practical implementation of an ML-model as a forecasting tool at HMS Networks are presented.

To achieve an accurate comparison between the ML-models and the HMS forecast, the data was filtered to only contain the SKUs that occur in both datasets. While increasing the comparability, the filtering reduced the number of SKUs by approximately 80%. The results show an increased performance according to the MAE + |bias| metric, for both boosting models. When measuring performance at the product category level, a 19% and 16.5% performance increase was seen from XGBoost and LightGBM respectively, compared to the HMS forecast. However, on the disaggregated level, the increased performance was only 6.2% and 4.9% for XGBoost and LightGBM respectively.

Regarding the implementation of machine learning in the demand forecasting process at HMS, there are both possibilities and limitations. As seen in the results, there is great potential in utilizing an ML-model as a tool for demand forecasting. By constantly gaining access to more sales data, the models should be able to improve over time and increase the accuracy of the generated forecasts. Furthermore, if implemented correctly, an ML-model would possibly reduce the manual labor of creating the forecasts.

However, there are limits to what the ML-models can forecast. The occurrence of one-time or customer-unique sales orders with large volumes skew the training data and affect the performance of the ML-models. Therefore, the ML-models should be viewed as baseline forecasts, that predict the behavior of the standard product mix. Combining the ability of the ML-models to identify trends and the human knowledge of unique orders and customer relationships could generate a further improved forecast.

To answer the third research question, there is great potential in applying machine learning as a tool for demand forecasting. In this thesis, several models have shown the ability to achieve more accurate results than the current forecast at HMS Networks. Furthermore, additional tweaking of the data and the models should lead to further improvements of the ML-models. However, the models are limited to achieving accurate results for SKUs with sufficient historical data. Thus, this thesis concludes that the combination of machine learning and human intelligence seems optimal, using ML-models as baseline forecasts and adjusting according to knowledge of the customers.

7.2 Contributions

The contributions of this thesis are divided into theoretical contributions to academia and practical contributions to HMS Networks.

The thesis shows the potential of ML-models as a tool for demand forecasting, confirming previous research and the increasing trend of including machine learning in the supply chain planning process, identified by Babai et al. (2023). Furthermore, the results of the thesis show the importance of both data representation and granularity when improving the performance of ML-based forecasts.

Firstly, the study demonstrates the importance of aligning the data representation with the limitations and capabilities of each model. The results indicate that the improved performance was primarily driven by better representations of the demand data, rather than through model selection alone. In particular, the study contributes to insights into the effects of first-order differencing and scaling for multi-series demand forecasting.

Secondly, the thesis also contributes to an improved understanding of the recursive forecasting methodology in a high granularity setting. The results show that weekly granularity may generate divergent recursive forecasts, while monthly granularity performs more stably. This highlights practical limitations of recursive forecasting when long forecasting horizons are combined with limited historical observations.

From a practical perspective, the thesis has contributed to HMS Networks by exploring different aspects of the development and implementation of a ML-based forecast as well as showing the potential to improve their forecast by implementing machine learning. Once again, the importance of data processing

is highlighted, rather than the choice of model. However, tree-based models are determined to offer promising performance. Regarding granularity, monthly forecasts on product category level perform the best in the context of HMS Networks. Worth noting, however, is that the temporal granularity has a larger impact on the forecasting performance than product aggregation.

Another contribution to HMS Networks is the understanding of the limitations of an ML-based forecast. The models should be continuously evaluated and adjusted to achieve greater performance. Furthermore, the variability in the sales data implies that an ML-model is insufficient to reliably replace the entire forecasting process. Instead it should be viewed as a baseline and adjusted based on knowledge of employees in the demand planning process.

7.3 Limitations and Future Research

In this section, limitations of the thesis are presented to understand potential improvements which would generate more reliable findings. Furthemore, recommendations for future research are presented which would deepen the knowledge within the field and further build upon the findings of this thesis.

One of the limitations of the thesis was that only a single case company, HMS Networks, was used when applying the research methodology. Although the results may be applicable to other scenarios, the generalizability could have been proven if multiple case companies were included in the research.

One of the aspects of the artifact development was the presence of stock-outs. Based on interviews, stock-outs is not a common issue for HMS Networks, and was therefore not taken into account when developing the artifacts. To increase the reliability of the research, it would be of interest to attain inventory data and determine the number of stock-out occurrences. Attaining and investigating the inventory data was not possible within the time scope of the thesis, but would be recommended for future research.

The ML-models forecasts were solely based on historical sales data, deeming them unable to include upcoming large orders or other factors that may affect the future demand. The current forecast at HMS Networks, on the other hand, utilizes the knowledge of employees in sales and demand planning. It would, therefore, be interesting to explore the potential of the ML-models when combining them with the knowledge of predictable changes that cannot be identified in historical data. This would not only potentially improve the performance of the models, but also make the results of the ML-models more comparable to the current forecast at HMS Networks. Furthermore, the historical occurrences of one-time, large-volume sales affect the forecasts of the ML-models. By identifying and excluding such outliers, the models would be able to, more accurately, predict a baseline demand forecast. For future research it would therefore be of interest to collaborate with the supply planning employees when generating a forecast. To be able to evaluate the results, a longer research horizon would therefore be needed.

The data filtering process is based on product master data from the end of 2025, which includes features such as life-cycle status and product status. Statuses are updated over time on the basis of the sales volume and age of each product. Since the ML-models base their predictions on historical sales data from 2021-2024, the filtering process may have included or excluded certain products of which the status has changed. Therefore, having access to product master data that includes all updates during the time period of interest would improve the filtering, and in turn possibly improve the forecasts.

In cycles 2 and 3, first-order differencing and the series mean were used to transform the demand data. The methods used were deemed appropriate, because of their underlying assumptions of data structure and simplicity. There are, however, other methods of detrending and scaling data that were not included in this thesis, due to time limitations. Future research could therefore include testing different approaches of data transformation to determine which methods are most appropriate to achieve an accurate forecast.

Furthermore, the models were trained with a limited number of time lags. This could have been increased by either introducing pre-pending zero-fill or by excluding product categories and product groups introduced at a later stage. As discussed, a greater number of time lags could allow models to capture relationships within the data with greater horizons. Additionally, this could reduce the divergence of weekly recursive forecasts.

Based on the review of literature, the recursive forecasting methodology was preferred over the direct forecasting methodology. However, from the granularity results, it was noted that the weekly forecasts grew unstable and diverged during the long forecasting horizons. Future research could hence benefit from comparing the two forecasting methodologies, giving further insights in to the methodologies' effect on forecast stability and performance.

Due to the scope and delimitations of the thesis, not all SKUs in HMS Networks' product portfolio were included. For future research it would therefore be appropriate to increase the scope and include a larger portion of the product portfolio. This would increase the understanding of how well ML-models can forecast products with varying life-cycle and product statuses.

References

Abolghasemi, M., Beh, E., Tarr, G., & Gerlach, R. (2020). Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & Industrial Engineering*, *142*, 106380. https://doi.org/10.1016/j.cie.2020.106380

Aliferis, C., & Simon, G. (2024). Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI [Series Title: Health Informatics]. In G. J. Simon & C. Aliferis (Eds.), *Artificial intelligence and machine learning in health care and medical sciences* (pp. 477–524). Springer International Publishing. https://doi.org/10.1007/978-3-031-39355-6_10

Altunkaynak, A., & Nigussie, T. A. (2018). Monthly water demand prediction using wavelet transform, first-order differencing and linear detrending techniques based on multilayer perceptron models. *Urban Water Journal*, *15*(2), 177–181. https://doi.org/10.1080/1573062X.2018.1424219

Babai, M. Z., Arampatzis, M., Hasni, M., Lolli, F., & Tsadiras, A. (2023). On the use of machine learning in supply chain management: A systematic review. *IMA Journal of Management Mathematics*, *36*(1), 21–49. https://doi.org/10.1093/imaman/dpae029

Berry, M. W., Mohamed, A., & Wah, Y. B. (Eds.). (2020). *Supervised and unsupervised learning for data science*. Springer. https://doi.org/10.1007/978-3-030-22475-2

Björklund, M. (2012). *Seminarieboken: Att skriva, presentera och opponera* (2. uppl). Studentlitteratur.

Bouhamed, O., Amayri, M., & Bouguila, N. (2022). Weakly supervised occupancy prediction using training data collected via interactive learning. *Sensors*, *22*(9), 3186. https://doi.org/10.3390/s22093186

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

De Ville, B. (2013). Decision trees. *WIREs Computational Statistics*, *5*(6), 448–455. https://doi.org/10.1002/wics.1278

Duffy, N., & Helmbold, D. (2002). Boosting methods for regression. *Machine Learning*, *47*(2), 153–200. https://doi.org/10.1023/A:1013685603443

Ertel, W., & Mast, F. (2025). *Introduction to artificial intelligence* (N. T. Black, Trans.; Third edition). Springer. https://doi.org/10.1007/978-3-658-43102-0

Fahrmeir, L., Kneib, T., Lang, S., & Marx, B. D. (2022). *Regression: Models, methods and applications* (2. 2nd ed. 2021). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-63882-8

Fouad, A. M., Wefki, H., & Kouta, H. (2025). AI-driven raw material demand forecasting: Towards project management practices. *Journal of Supply Chain Management Science*, *6*(1). https://doi.org/10.59490/jscms.2025.8136

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5). https://doi.org/10.1214/aos/1013203451

Gunay, D. (2023, September 11). *Random forest* [Medium]. Retrieved March 17, 2026, from https://medium.com/@denizgunay/random-forest-af5bde5d7e1e

Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, *61*(4), 5–14. https://doi.org/10.1177/0008125619864925

Hammam, I. M., El-Kharbotly, A. K., & Sadek, Y. M. (2025). Adaptive demand forecasting framework with weighted ensemble of regression and machine learning models along life cycle variability. *Scientific Reports*, *15*(1), 38482. https://doi.org/10.1038/s41598-025-23352-w

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning*. Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Hatakeyama-Sato, K., & Oyaizu, K. (2021). Generative models for extrapolation prediction in materials informatics. *ACS Omega*, *6*(22), 14566–14574. https://doi.org/10.1021/acsomega.1c01716

Henckaerts, R., Côté, M.-P., Antonio, K., & Verbelen, R. (2019). Boosting insights in insurance tariff plans with tree-based machine learning methods [Version Number: 3]. https://doi.org/10.48550/ARXIV.1904.10890

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research1. *MIS Quarterly*, *28*(1), 75–106. https://doi.org/10.2307/25148625

Hyndman, R. J., & Athanasopoulos, G. (2014). *Forecasting: Principles and practice; [a comprehensive introduction to the latest forecasting methods using r; learn to improve your forecast accuracy using dozens of real data examples]*. Otexts.

Ingle, C., Bakliwal, D., Jain, J., Singh, P., Kale, P., & Chhajed, V. (2021). Demand forecasting : Literature review on various methodologies. *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–7. https://doi.org/10.1109/ICCCNT51525.2021.9580139

Johannesson, P., & Perjons, E. (2021). Research strategies and methods. In *An introduction to design science* (pp. 41–75). Springer International Publishing. https://doi.org/10.1007/978-3-030-78132-3_3

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *30*. https://doi.org/doi/10.5555/3294996.3295074

Knauss, E. (2021). Constructive master's thesis work in industry: Guidelines for applying design science research. *2021 IEEE/ACM 43rd International*

Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET), 110–121. https://doi.org/10.1109/ICSE-SEET52601.2021.00021

Kolassa, S., Rostami-Tabar, B., & Siemsen, E. (2023, July 21). *Demand forecasting for executives and professionals* (1st ed.). Chapman; Hall/CRC. https://doi.org/10.1201/9781003399599

Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling*. Springer New York. https://doi.org/10.1007/978-1-4614-6849-3

Kumar Dubey, A., Kumar, A., García-Díaz, V., Kumar Sharma, A., & Kanhaiya, K. (2021). Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustainable Energy Technologies and Assessments*, *47*, 101474. https://doi.org/10.1016/j.seta.2021.101474

Leokhin, Y., Fatkhulin, T., Zanegin, A., & Rakhmatova, A. (2025). Researching the efficiency of machine learning methods used in forecasting demand for certain types of goods. *2025 Systems of Signals Generating and Processing in the Field of on Board Communications*, 1–8. https://doi.org/10.1109/IEEECONF64229.2025.10948113

Liapis, C. M., Karanikola, A., & Kotsiantis, S. (2021). A multi-method survey on the use of sentiment analysis in multivariate financial time series forecasting. *Entropy*, *23*(12), 1603. https://doi.org/10.3390/e23121603

Mehdiyev, N., Enke, D., Fettke, P., & Loos, P. (2016). Evaluating forecasting methods by considering different accuracy measures. *Procedia Computer Science*, *95*, 264–271. https://doi.org/10.1016/j.procs.2016.09.332

Montero-Manso, P., & Hyndman, R. J. (2021). Principles and algorithms for forecasting groups of time series: Locality and globality. *International Journal of Forecasting*, *37*(4), 1632–1653. https://doi.org/10.1016/j.ijforecast.2021.03.004

Myers, M. D., & Venable, J. R. (2014). A set of ethical principles for design science research in information systems. *Information & Management*, *51*(6), 801–809. https://doi.org/10.1016/j.im.2014.01.002

Nagadevi, S., Abirami, G., Vidhya, R., Selvakumar, S., & Vijayalakshmi, C. (2025). XGBoost-based demand forecasting in supply chain management using machine learning algorithm. *International Journal of Interactive Mobile Technologies (iJIM)*, *19*(18), 161–174. https://doi.org/10.3991/ijim.v19i18.57253

Nasteski, V. (2017). An overview of the supervised machine learning methods. *HORIZONS.B*, *4*, 51–62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05

Özkurt, C. (2024). Enhancing financial decision-making: Predictive modeling for personal loan eligibility with gradient boosting, XGBoost, and AdaBoost. *Information Technology in Economics and Business*, *1*(1), 7–13. https://doi.org/10.69882/adba.iteb.2024072

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77. https://doi.org/10.2753/MIS0742-1222240302

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Ben Taieb, S., Bergmeir, C., Bessa, R. J., Bijak, J., Boylan, J. E., Browell, J., Carnevale, C., Castle, J. L., Cirillo, P., Clements, M. P., Cordeiro, C., Cyrino Oliveira, F. L., De Baets, S., Dokumentov, A., ... Ziel, F. (2022). Forecasting: Theory and practice. *International Journal of Forecasting*, *38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Ranawana, R., & Karunananda, A. S. (2021). An agile software development life cycle model for machine learning application development. *2021 5th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI)*, 1–6. https://doi.org/10.1109/SLAAI-ICAI54477.2021.9664736

Rowley, J., & Slack, F. (2004). Conducting a literature review. *Management Research News*, *27*(6), 31–39. https://doi.org/10.1108/01409170410784185

Shrivastava, S. (2020, January 14). *Cross validation in time series* [Medium]. Retrieved April 23, 2026, from https://medium.com/@soumyachess1496/cross-validation-in-time-series-566ae4981ce4

Svetunkov, I. (2024, May 25). *Recursive vs direct forecasting strategy* [Open forecasting]. Retrieved February 13, 2026, from https://openforecast.org/2024/05/25/recursive-vs-direct-forecasting-strategy/

Taieb, S. B., & Hyndman, R. J. (2012). Recursive and direct multi-step forecasting: The best of both worlds. (19). https://www.academia.edu/2767618/Recursive_and_direct_multi_step_forecasting_the_best_of_both_worlds

Vandeput, N. (2023). *Demand forecasting best practices*. Manning.

Verdonck, T., Baesens, B., Óskarsdóttir, M., & Vanden Broucke, S. (2024). Special issue on feature engineering editorial. *Machine Learning*, *113*(7), 3917–3928. https://doi.org/10.1007/s10994-021-06042-2

Vom Brocke, J., Hevner, A., & Maedche, A. (2020). Introduction to design science research [Series Title: Progress in IS]. In J. Vom Brocke, A. Hevner, & A. Maedche (Eds.), *Design science research. cases* (pp. 1–13). Springer International Publishing. https://doi.org/10.1007/978-3-030-46781-4_1

Voyant, C., Despotovic, M., Garcia-Gutierrez, L., Silva, R. A. E., Lauret, P., Soubdhan, T., & Bailek, N. (2025). NICE k metrics: Unified and multi-dimensional framework for evaluating deterministic solar forecasting accuracy. *Sustainable Energy Technologies and Assessments*, *83*, 104588. https://doi.org/10.1016/j.seta.2025.104588

Wandre, R., Dhalkari, V., Ahirekar, S., Hol, S., & Patel, N. (2025). Comparative analysis of demand forecasting models for medical drugs: XGBoost, random forest, and prophet. *2025 4th International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 1211–1215. https://doi.org/10.1109/ICAAIC64647.2025.11331141

Wang, C.-C. (2025). Dynamic dual-phase forecasting model for new product demand using machine learning and statistical control. *Mathematics*, *13*(10), 1613. https://doi.org/10.3390/math13101613

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, *1168*, 022022. https://doi.org/10.1088/1742-6596/1168/2/022022

Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble machine learning: Methods and applications*. Springer New York. https://doi.org/10.1007/978-1-4419-9326-7

Zhou, Z.-H. (2021). *Machine learning* (S. Liu, Trans.). Springer.

A Semi-Structured Interview

The following list shows the questions asked during the semi-structured interview with the market manager.

1. Introduction
    a) Can we record the interview?
    b) *Explaining the thesis.*
2. About the Interviewee
    a) Our understanding is that you are a Market Manager at the Division of Industrial Data Solutions, is this accurate?
    b) Would you like to tell us about your role at HMS Networks?
    c) What are your main responsibilites in this role?
    d) What geographical areas do you mainly work with?
3. Demand Forecasting
    a) *Explaining our understanding of the demand forecasting process at HMS Networks.*
    b) Does this correspond to your understanding of the demand forecasting process? If not, what differs?
4. Market Overview
    a) Have you noticed any major changes in demand over the last few years (e.g. due to Covid-19)?
    b) Are there any particular market trends that influece demand of your products?
    c) When it comes to new competition, what is your strategy to gain or keep market shares?
    d) What does HMS Networks focus on to ensure market progression compared to the competition?
5. Customer Behavior
    a) What factors do you think influence customers' decisions to buy your products?
    b) Are there particular customer needs or trends that affect demand?
6. Promotion and Sales Activities
    a) What types of promotions or campaigns are most common in your market?

b) Do you notice demand increasing before, during or after promotions/campaigns?

c) Do customers react strongly to price changes or campaigns?

d) Do campaigns by competition affect your sales?

e) How often do you experience stock-outs of a product?

f) How often would a stock-out lead to losing a customer?

g) Which sales activities have the biggest impact on demand?

h) Do you affect the demand forecast in your day-to-day work, or vice versa?

B Metrics of Different Granularity Models

Table 22 shows the resulting validation and test scores of each ML-model based on the four variations of time clumping and product aggregation.

		Val. MAE	MAE	MAPE	RMSE	MAE + |bias|
Group & Month	DecisionTree	345,9	322,4	118,5	771,1	430,5
	RandomForest	341,5	261,2	113,9	646,2	267,1
	LightGBM	382,6	261,4	97,9	696,3	338,2
	XGBoost	343,9	267,1	125,0	702,4	300,7
Group & Week	DecisionTree	153,3	167,2	263,6	402,7	202,8
	RandomForest	203,7	428,7	584,2	1136,9	840,5
	LightGBM	153,9	140,4	251,7	369,7	201,9
	XGBoost	142,6	264,7	385,4	729,4	489,6
Category & Week	DecisionTree	260,7	325,4	136,5	663,9	368,5
	RandomForest	339,7	682,0	320,0	1477,4	1321,4
	LightGBM	288,1	337,4	204,7	707,3	576,4
	XGBoost	260,6	345,1	186,9	747,9	581,5
Category & Month	DecisionTree	587,7	684,6	148,7	1530,7	948,3
	RandomForest	592,6	600,4	110,4	1368,0	630,2
	LightGBM	588,7	524,8	85,1	1213,4	646,5
	XGBoost	587,3	519,4	95,0	1194,2	646,0

Table 22: Test metrics for various granularity settings.

C Error Metrics on SKU Level

Table 23 shows the MAE, bias and MAE + |bias| metrics for each SKU averaged on product category level for the forecast created by HMS Networks.

Category	**MAE**	**Bias**	**MAE + |bias|**
Category A	14,3	4,1	22,0
Category B	5,3	0,2	8,2
Category C	52,7	-9,5	87,8
Category D	86,3	-52,8	159,2
Category E	21,0	-12,5	35,2
Category F	14,5	-4,3	24,4
Category G	71,3	0,4	88,4
Category H	45,3	6,3	65,6
Category I	17,0	6,7	26,2
Category J	33,0	7,0	52,8
Category K	82,0	-26,8	140,1
Category L	37,0	-23,6	68,0
Category M	30,0	-13,3	50,9
Category N	14,6	-8,6	23,5
Category O	21,7	-21,6	43,3
Grand Total	**45,6**	**-14,8**	**77,2**

Table 23: Mean Error Metrics of HMS forecast on SKU level per Category.

Table 24 shows the MAE, bias and MAE + |bias| metrics for each SKU averaged on product category level for the forecast generated by the XGBoost model.

Category	**MAE**	**Bias**	**MAE + |bias|**
Category A	13,0	5,0	19,6
Category B	4,7	2,3	7,2
Category C	57,4	22,0	93,5
Category D	76,6	-6,9	124,4
Category E	20,9	-0,9	33,0
Category F	10,7	-0,4	14,5
Category G	71,2	17,6	99,0
Category H	67,9	42,6	113,0
Category I	14,0	8,0	22,7
Category J	26,3	6,2	36,1
Category K	73,2	-29,9	126,0
Category L	36,4	-25,9	64,3
Category M	27,9	-3,5	47,7
Category N	12,2	-0,6	15,4
Category O	30,7	-9,9	49,3
Grand Total	**44,4**	**2,3**	**72,4**

Table 24: Mean Error Metrics of the XGBoost model on SKU level per Category.

D Error Metrics by Category

In the following tables the results of the ML-models and the HMS forecast are per product category. The forecasts are based on month and product category granularity, and the metrics are calculated on the category level.

Catgory	**Decision Tree**	**Random Forest**	**LightGBM**	**XGBoost**	**HMS**
Category A	338,7	276,1	223,0	195,4	241,1
Category B	31,8	20,3	13,0	18,3	13,5
Category C	4148,4	3330,1	2140,4	1804,9	865,6
Category D	1799,6	1410,6	1369,3	1287,9	2956,3
Category E	312,1	161,3	128,5	117,4	438,3
Category F	163,2	132,8	128,9	131,1	192,6
Category G	561,3	491,9	392,8	464,9	457,8
Category H	1760,9	960,7	956,8	971,9	413,0
Category I	200,9	142,5	145,7	103,6	125,6
Category J	193,4	157,1	142,8	138,1	151,2
Category K	1030,8	1156,6	1203,6	1344,5	1804,3
Category L	263,5	265,6	269,6	279,8	276,6
Category M	457,0	428,9	437,2	328,6	738,9
Category N	29,7	29,1	29,0	26,1	40,8
Category O	98,0	83,8	72,1	77,1	65,1
Average	**759,3**	**603,2**	**510,2**	**486,0**	**585,4**

Table 25: MAE error metric on category level, for filtered SKUs.

Catgory	**Decision Tree**	**Random Forest**	**LightGBM**	**XGBoost**	**HMS**
Category A	332,3	273,4	212,4	190,4	154,4
Category B	30,4	18,0	8,5	15,8	1,7
Category C	4056,4	3330,1	1792,3	1782,7	-772,3
Category D	553,1	-466,6	-631,2	-386,9	-2956,3
Category E	245,4	53,8	25,4	-32,1	-438,3
Category F	124,4	24,2	-24,3	-11,2	-134,6
Category G	510,8	139,0	110,8	193,6	4,7
Category H	1545,7	916,9	929,2	809,6	119,7
Category I	200,9	142,5	145,7	103,6	87,6
Category J	152,0	105,5	64,4	55,6	63,0
Category K	-753,7	-1085,7	-1082,9	-1344,4	-1205,1
Category L	-167,7	-222,9	-242,8	-259,0	-235,9
Category M	204,1	-300,4	-98,5	-158,4	-597,3
Category N	16,4	-11,8	-7,3	-2,4	-34,4
Category O	4,3	-19,3	-38,1	-29,7	-64,8
Average	**470,3**	**193,1**	**77,6**	**61,8**	**-400,5**

Table 26: Bias error metric on category level, for filtered SKUs.

Catgory	**Decision Tree**	**Random Forest**	**LightGBM**	**XGBoost**	**HMS**
Category A	670,9	549,5	435,3	385,8	395,5
Category B	62,2	38,3	21,4	34,1	15,2
Category C	8204,8	6660,3	3932,6	3587,6	1637,8
Category D	2352,7	1877,1	2000,4	1674,8	5912,7
Category E	557,5	215,1	153,9	149,5	876,7
Category F	287,6	157,0	153,2	142,4	327,2
Category G	1072,1	630,9	503,7	658,5	462,5
Category H	3306,6	1877,6	1886,0	1781,5	532,7
Category I	401,7	285,0	291,4	207,2	213,2
Category J	345,3	262,5	207,1	193,8	214,2
Category K	1784,5	2242,3	2286,4	2688,9	3009,3
Category L	431,2	488,5	512,4	538,8	512,5
Category M	661,1	729,3	535,7	487,0	1336,2
Category N	46,0	41,0	36,4	28,5	75,2
Category O	102,3	103,1	110,1	106,9	129,8
Average	**1352,4**	**1077,2**	**871,1**	**844,3**	**1043,4**

Table 27: MAE + |bias| error metric on category level, for filtered SKUs.