Aalto University
School of Science

Master's Programme in Mathematics and Operations Research

Forecasting Long-term Food Demand in Humanitarian Logistics

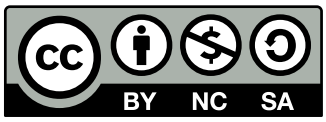
A United Nations World Food Programme Case Study

Johan Lindell

Master's Thesis
2025

© 2025

This work is licensed under a Creative Commons "Attribution-NonCommercial-ShareAlike 4.0 International" license.



Aalto University
School of Science

Author Johan Lindell

Title Forecasting Long-term Food Demand in Humanitarian Logistics — A United Nations World Food Programme Case Study

Degree programme Mathematics and Operations Research

Major Systems and Operations Research

Supervisor Prof. Ahti Salo

Advisors M.Sc. Diego Roa, M.Sc. Juuso Heinonen

Collaborative partner Netlight, UN World Food Programme

Date 18 July 2025 **Number of pages** 80 **Language** English

Abstract

Increases in conflict, economic instability, and accelerating climate change, coupled with a decrease in humanitarian funding is causing a rise in global hunger. Consequently, more efficient humanitarian supply chains are needed in order to decrease costs and ultimately deliver more aid to more people. Demand forecasting is crucial in efficient supply chain management, ultimately enabling better supply planning and inventory optimization. However, literature on long-term demand forecasting within the humanitarian sector is scarce. This thesis aims to fill this gap by analyzing various forecasting methods for predicting food commodity demand with a horizon of 12 months in collaboration with the United Nations World Food Programme. In this thesis we implemented and evaluated naive, statistical, judgmental, and machine learning methods. Moreover, for the machine learning methods, various feature engineering methods and exogenous data points were tested. Finally, since the uncertainty of the forecast is key to informed supply chain management decision-making, a quantile forecasting model based on the point-forecasting model was proposed and evaluated.

This thesis found statistically significant accuracy gains in LightGBM models when compared to naive and statistical models such as moving average, Auto-ARIMA, and Holt-Winters exponential smoothing models. The machine learning models were tuned using cross-validation, and testing was performed in an expanding window backtesting fashion to ensure the robustness and stability of the models. We found that a LightGBM direct model with normalized data had the best performance as compared to all other models. Quantile forecasting with LightGBM predicted quantiles with less error compared to models that assumed a Gaussian distribution, however with the upper quantiles being unstable. These findings demonstrate that machine learning methods can increase the accuracy of humanitarian demand forecasting as compared to naive and statistical approaches.

Keywords Humanitarian logistics, Demand forecasting, LightGBM, Grouped time series, Probabilistic forecasting, Judgmental forecasting

Aalto-universitetet
Högskolan för
teknikvetenskaper

Författare Johan Lindell

Titel Prognostisering av långsiktig livsmedelsefterfrågan inom humanitär logistik — En FN World Food Programme fallstudie

Utbildningsprogram Mathematics and Operations Research

Huvudämne Systems and Operations Research

Övervakare Prof. Ahti Salo

Handledare M.Sc. Diego Roa, M.Sc. Juuso Heinonen

Samarbetspartner Netlight, FN World Food Programme

Datum 18 Juli 2025 **Sidantal** 80 **Språk** engelska

Sammandrag

Ökande konflikter, ekonomisk instabilitet och accelererande klimatförändring med en minskande humanitär finansiering leder till en ökning i global hunger. Följaktligen behövs effektivare humanitära leveranskedjor för att minska kostnader och i slutändan leverera mer bistånd till fler människor i nöd. Efterfrågeprognoser är viktiga för en effektiv hantering av leveranskedjor, vilket i slutändan möjliggör bättre leveransplanering och lageroptimering. Litteratur inom långsiktiga efterfrågeprognoser i den humanitära sektorn saknas. Syftet med denna avhandling är att utöka litteraturen på efterfrågeprognoser i den humanitära sektorn. Detta gjordes genom att analysera olika prognosmetoder för att förutspå efterfrågan av livsmedelsråvaror för 12 månader framåt i samarbete med Förenta Nationernas World Food Programme. För denna analys implementerades olika naiva, statistiska, bedömningsbaserade och maskininlärningsmetoder (ML) för att evalueras. För ML metoderna användes olika exogena datapunkter och data transformationer för att öka noggrannheten av prognoserna. Osäkerheten i prognosen är avgörande när beslut fattas baserat på dem, särskilt i volatila miljöer. Därför har maskininlärningsmodellerna utvidgats för att även kunna uppskatta prognosernas kvantiler.

Denna avhandling fann att LightGBM ML-metoder hade statistiskt signifikant högre noggrannhet än statistiska, naiva och bedömningsbaserade metoder. ML-metoderna var finjusterade med korsvalidering och testades vid flera tidpunkter för att säkerställa modellens robusthet och stabilitet. LightGBM med en direkt prognosmetod och normaliserad data presterade bättre än alla andra modeller. Vi fann att kvantilprognos med LightGBM presterade bättre än metoder med antagande om gaussisk distribution, dock fann vi de högre kvantilprognoserna ostabila. Dessa resultat visar att ML-metoder kan öka noggrannheten i prognoser inom humanitär livsmedelsråvarefterfrågan jämfört med statistiska och naiva metoder.

Nyckelord Humanitär logistik, Efterfrågeprognoser, , LightGBM, Grupperade tidsserier, Probabilistisk prognostisering, Bedömningsbaserad prognostisering

Preface and Acknowledgments

This thesis was conducted in collaboration with the World Food Programme's (WFP) Supply Chain Planning & Optimization Unit. I extend my gratitude to Eric Combette and Ricardo Marques for their technical and domain expertise regarding this thesis.

I am deeply appreciative of the support received from Netlight and wish to thank everyone involved for enabling this collaborative endeavor. Special thanks to Anna Routti and Moritz Tränkner-Tuborgh for fostering the partnership within the NGO space, particularly with WFP. My thanks also go to Gunnar Risting for initiating the collaboration with the SCOUT team, to David Rochholz for his consistently wise counsel, and to Katariina Kesseli for her mentorship throughout this period.

I would like to express my gratitude to my supervisor, Ahti Salo, and my advisors, Juuso Heinonen and Diego Roa, for support and guidance on both the technical and academic aspects of this thesis.

Finally, a special thanks to my family, and especially to Emmi, for their consistent support and proofreading during what has been an intensive phase of my life.

This thesis was written during a significant period within the humanitarian sector. Large funding cuts and increases in conflict have deepened the need for humanitarian assistance. It is my hope that this work will contribute, even in a small way, to improving the effectiveness of humanitarian response.

Helsinki, 18 July 2025

Johan Lindell

Contents

Symbols and abbreviations

Symbols

Symbols that remain constant during the entire document are introduced here, other symbols are specified within their given context.

ϵ	Residual
y	Explanatory/target variable
x	Exogenous variable/feature
L	Loss function
h	Forecast horizon
t	Time
T	Last time point of training set
p	Number of features
τ	Quantile
Q	Set of quantiles
S	Set of segments (country, commodity, activity type) where $s \in S$

Abbreviations

ARIMA	AutoRegressive integrated moving average
CO	Country office
ES	Exponential smoothing
GBDT	Gradient boosting decision tree
HO	Humanitarian organization
IASC	Inter-Agency Standing Committee
IP	Implementation plans
IPC	Integrated food security phase classification
LightGBM	Light gradient boosting machine
MA	Moving average
MAE	Mean absolute error
ML	Machine learning
MSF-OCA	Operational Center Amsterdam of Medecins Sans Frontieres
PL	Pinball loss
RMSE	Root mean squared error
UN	United Nations
WFP	World Food Programme

1 Introduction

Demand forecasting is critical when designing an optimal supply chain within humanitarian operations [1]. Leveraging methods such as prepositioning and optimal commodity procurement can have substantial cost savings for humanitarian organizations [2]. These strategies can be improved via demand time series forecasting [3]. However, the literature on demand forecasting within humanitarian logistics is limited [1]. To address this gap, this thesis will develop and naive, statistical and machine learning (ML) models for forecasting food commodity demand, validated through a United Nations (UN) World Food Programme (WFP) case study.

1.1 Motivation

In 2015, the UN Sustainable Development Goal 2 was established, with the goal of ending world hunger by 2030 [4]. However, progress has been slow and stagnated since the Covid-19 crisis [5]. According to UN organizations [5], around 733 million people experienced hunger in 2023, which is approximately 150 million more than in 2019. The impact of these crises are profound, encompassing not only the immediate loss of life but also long-term health implications that extend across generations [6]. Recognizing these costs, humanitarian organizations (HOs) have provided aid to areas affected by crises [5], following the humanitarian principles of humanity, neutrality, impartiality, and independence [7, p. 20], [8]. Upholding these principles, particularly humanity and impartiality, depends on the logistical capability to deliver aid consistently to those most in need. This requires robust supply chains where service levels are high and stock is managed effectively to prevent interruptions [7, p. 20-25]. Large HOs rely on complex supply chains to deliver timely aid, the aid delivery process is known as humanitarian logistics [7].

Laan, Brito, Fenema, *et al.* [9] defines humanitarian logistics as "the process of planning, implementing and controlling the flow and storage of goods and materials as well as related information, from point of origin to point of emergency, for the purpose of meeting the end beneficiary's requirements". Additionally, Holguín-Veras, Pérez, Jaller, *et al.* [10] expanded the definition by optimizing the movement of goods to ensure "the greatest good for the greatest number of people". Humanitarian supply chains need to be agile and adaptable, with the ability to respond quickly and efficiently to volatile needs in different parts of the world [7, p. 7-8]. This means that supply planning and inventory optimization are crucial for minimizing costs and delivering aid quickly. A commonly used strategy in supply planning involves procuring and storing commodities in advance, known as prepositioning, which enables shorter delivery times and purchasing at more favorable prices [2]. Choosing optimal prepositioning volume requires forecasts of future demand as overstocking can lead to high inventory costs and risk of waste, tying up capital that could be used elsewhere [11]. Conversely, under-stocking not only leads to higher last-minute procurement costs and slower deliveries but, more critically, it directly threatens service levels. A stock-out can mean a disruption in aid, forcing beneficiaries to go without essential food and undermining the core mission of the organization [12].

While the available forecasting literature in humanitarian logistics focuses almost exclusively on short-term, post-disaster predictions, longer-term operations require longer-term planning. This need for longer-term planning and, therefore, longer-term forecasting is driven by multiple reasons, two important ones being optimal procurement prices and geographical constraints. First, significant seasonal price variability allows for strategic procurement. For example, a Gilbert, Christiaensen, and Kaminski [13] analysis of 13 commodities in 7 African countries revealed seasonal price gaps of up to 68%, highlighting the cost-saving potential of purchasing goods when prices are low. Moreover, for more protracted crises it is generally more cost efficient to allow for longer lead-times. Efficient procurement has large cost-saving potential since HOs procurement costs account for approximately 60% of the annual budget [2]. Second, logistical constraints can render prepositioning essential for ensuring continuous aid. For instance, the eastern part of South Sudan is inaccessible during the rainy season, requiring food supplies to be placed well in advance (see Figure 1) [14].[1] Moreover, the prepositioning of supplies shortens lead-times during crises, ensuring faster distribution of aid.

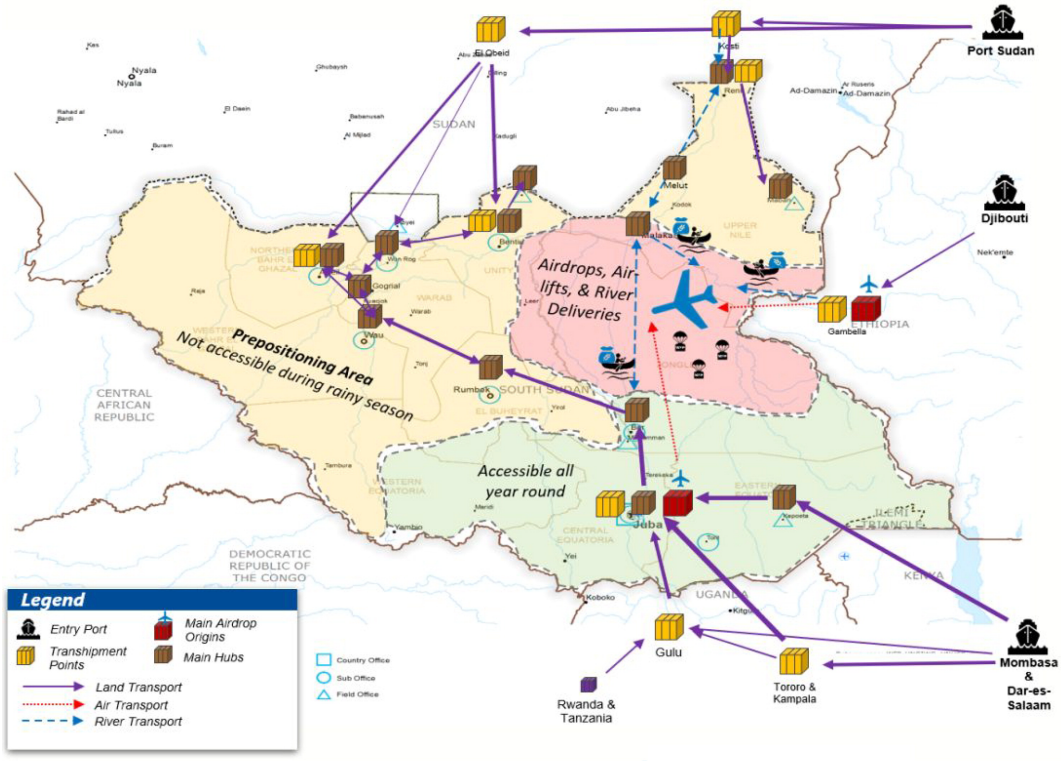


Figure 1: Overview of the UN World Food Programme supply network in South Sudan, highlighting the need for prepositioning [14].

[1] Port Sudan is currently not in use for deliveries to South Sudan due to conflict, therefore this figure is illustrative and does not reflect the current supply chains.

Demand forecasting within the humanitarian sector is limited; however, there is plenty of literature in the for-profit world [15]. Demand in the for-profit sector is consumer-driven, in contrast humanitarian demand, is primarily donor-driven and supply-constrained. As a result, humanitarian demand is often unknown until a crisis unfolds, changes rapidly with sudden peaks, and is inherently more volatile, presenting a fundamentally different forecasting problem [7, p. 9-16]. Therefore, to improve accuracy, demand forecasts within HOs should incorporate relevant exogenous data such as donor funding, climate models, and macroeconomic indicators, an approach shown to increase accuracy in forecasting literature [16]–[18].

Large HOs such as WFP have developed data platforms, these are systems with the purpose of storing and analyzing large amounts of data [14]. The development of such platforms enable data driven analysis methods such as ML-based forecasting. ML-based forecasting is a modeling approach used to identify complex temporal relationships in data [15]. This approach helps generate forecasts without requiring the specification of the interactions between input variables, which enables fast development and comparison of different models [15], [16]. Moreover, ML methods can leverage data in a cross-learning fashion, being able to draw trends and patterns from different timeseries increasing accuracy [18]. Despite the advantages of ML, traditional statistical models such as auto-regressive integrated moving average (ARIMA) [19] can still outperform more complex models [15]. Therefore, traditional statistical models should be used as benchmarks when comparing the performance of ML models [15].

Among the rise of ML-based models, gradient boosting decision trees (GBDTs) have seen promise in complex demand forecasting tasks [15], [18]. GBDTs achieve high performance on complex data while generally being faster to develop and less computationally intensive than other ML models like long-short term memory (LSTM) [17] or transformer based models like TFT [20]. Notably, recent studies, including the M5-forecasting competition [18], have shown that Light gradient boosting machine (LightGBM) [21], a contemporary version of GBDT, compares favorably to various other models and is recognized as a valid approach for addressing complex forecasting tasks.

Alongside quantitative forecasting techniques (e.g., naive, statistical, and machine learning models), human expertise can enhance predictive accuracy [15]. This approach, formally known as 'judgmental forecasting', relies on human expertise and can be utilized as a standalone method for inferring future demand. Moreover, hybrid methods that integrate quantitative models with human judgment are common, and these have shown to yield accuracy improvements when compared to purely judgmental or quantitative approaches in some cases [22].

This thesis aims to develop and analyze demand forecasting models for large HOs, with the UN WFP serving as a case study. The focus will be on predicting the demand for various commodities, countries, and distribution types. This presents a hierarchical forecasting task, requiring the prediction of multiple time series within a structured hierarchy. This timeseries structure, similar to the M5-competition [18], has the potential of utilizing cross-learning when developing forecasting models which is one of the motivations of using ML-based methods. Furthermore, research in both humanitarian and commercial sectors has demonstrated the benefits of incor-

porating exogenous data in forecasting tasks [16]–[18]. Moreover, by utilizing WFP implementation plans as judgmental forecasts and data features for the ML models, a combination of quantitative and judgmental forecasts can be produced. Consequently, this thesis will employ ML-based methods, particularly LightGBM, to predict demand, aiming to improve strategic prepositioning, optimize stock management, and ultimately enhance the service levels provided to beneficiaries. To ensure a robust evaluation of these methods, their performance will be benchmarked against established naive and statistical forecasting models.

1.2 Research Questions

The goal of this thesis is to develop and study models that improve long-term demand forecasting for a large HOs. More specifically, implementing different versions of LightGBM and compare these to naive and statistical models fitted on historical data. The models developed in this thesis should have the capability of incorporating exogenous data, since there is strong evidence that funding, climate and other external factors correlate with humanitarian operations [16], [17]. There is also strong evidence that points to the improved accuracy with exogenous data from the for-profit sector [18]. However, to keep the scope of this thesis, the exogenous variables will be few and will serve as a proof of concept for future development. To further keep a sound scope, this thesis will not evaluate the performance of LightGBM to other state of the art models such as LSTM [17] or TFT [20], but seeks to give insight into the use of LightGBM in this context and different implementations of LightGBM. This is not to say that the choice of using LightGBM is arbitrary, the motivation for this model will be further discussed in Section 4.

Ultimately, demand forecasting models are used in decision making, therefore the stability and accuracy of the predictions are important. In light of this, this thesis will study the reliability and explainability of the models using SHAP values [23] and backtesting [15]. For robust and well-informed decisions to be made based on forecasting, the uncertainty of the forecast is crucial [24], especially in volatile time series such as humanitarian demand. Therefore, this thesis will implement probabilistic forecasts with LightGBM models by predicting quantiles. More specifically, this thesis seeks to answer the following research questions:

1. How accurately can baseline forecasting methods, including naive, statistical, and judgmental models, predict funding-constrained humanitarian demand?

2. Can LightGBM models improve the accuracy as compared to the baseline methods and what variation of the chosen model is the most accurate based on the chosen evaluation metrics?

3. How reliable and interpretable are the LightGBM model predictions, as assessed through cross-validation and SHAP analysis?

4. How effectively can quantile forecasts from the LightGBM models characterize the uncertainty associated with constrained humanitarian demand?

This thesis is developed within the operational context of the WFP and aims to support and enhance existing demand forecasting efforts. By presenting a proof of concept model as a case study, the work provides a foundation for practical implementation and highlights opportunities for further development. The findings have the potential to improve the efficiency and accuracy of WFP's current forecasting processes, ultimately contributing to more effective operational planning. Moreover, the goal is to provide a study from which the HO or other organizations can further iterate on. In addition to maintaining accuracy, the model must also be adaptable by including or omitting external data. It should accommodate the inclusion or exclusion of time series, as new commodities might be introduced, and old ones decommissioned or operations paused. Moreover, this model is expected to be retrained and produce new forecasts continuously so the most up to date forecasts can be used in supply planning. In the future, the model should support the development of a probabilistic framework. Incorporating these features will allow for a flexible model that can also model the uncertainty of the forecasts with the goal of aiding supply chain planning within humanitarian logistics.

1.3 Structure

As a research framework, Design Science Research Methodology (DSRM) within information science will be used [25]. DSRM is composed of six activities from problem definition and motivation, model design and finally communication of findings, these activities will guide the general structure of this thesis. The thesis will be structured as follows. Section 1 provides the motivation and defines the research problem and questions. Section 2 provides an overview of humanitarian logistics and subsequently reviews the existing literature within demand forecasting. In Section 3, the theory and methods used in the study will be introduced. In Section 4, the case study will be introduced and the implementation formulated. In Section 5, the results from the case study will be shown , as well as compared with existing literature. In Section 6, this thesis's results will be discussed, and potential future research directions will be explored. Lastly, Section 7 will give a summary of the research problem, the methodologies used, and the results of this thesis.

2 Background

This section introduces the concept of humanitarian logistics, with a specific focus on food aid. It then reviews the relevant literature on forecasting in the humanitarian sector and contrasts these findings with research from the for-profit sector.

2.1 Humanitarian Logistics

Ultimately, logistics is concerned with bringing items from point A to point B. In the humanitarian context, this usually means bringing aid from a source (warehouse, factory, or farm) to beneficiaries in need [7]. In this thesis, the terms aid, commodities, or items specifically refer to food aid unless noted otherwise. It is not enough that food is delivered, since this can be done in many ways. When referring to humanitarian logistics these must adhere to the humanitarian principles, humanity, neutrality, impartiality, and independence [7, p. 20], [8]. The principle of humanity includes addressing human suffering wherever it is found, especially for the most vulnerable. Neutrality entails offering help without siding with parties in disputes or armed conflicts. Impartiality ensures that aid is provided without discrimination, based solely on the need. Independence requires that such actions remain separate from any political, economic, or military objectives.

Humanitarian logistics seeks to alleviate suffering in humanitarian crises. These crises can be divided into sudden and slow-onset crises [1]. The objective of humanitarian operations is to mitigate the damage of such crises, this is called the disaster management cycle, which can be split into four phases: mitigation, preparedness, response and recovery [11]. The phases before a crisis, including mitigation and preparedness, are essential to reduce the damage from a crisis. The best results in preparation and mitigation are achieved with slow-onset crises, as these can be anticipated, allowing meaningful preparations to be implemented [1]. However, with sudden-onset crises such as earthquakes which are near-impossible to predict, preparedness is difficult and mitigation sometimes impossible [1]. With large sudden-onset crises, sometimes the preparedness is not sufficient to meet the needs of the affected, this is when a large response quickly is needed. These quick and large responses are also known as Humanitarian System-Wide Scale-Up Activation (or Scale-Ups) are needed [26]. Scale-Ups do not just affect one HO but are coordinated between multiple relevant organizations by the Inter-Agency Standing Committee (IASC) [26]. When analyzing demand curves, scale-ups are sudden and the magnitude of demand large. This brings difficulties in forecasting and traditional methods are usually not sufficient [27]. In scale-ups Ouchtar, t'Serstevens, and Rahman [27] recommend using different methods for forecasting such as scenario-based planning. Forecasting and modeling crises is essential however a key problem is the lack of structured and up to date data.

2.2 Food Aid

There are different types of aid, for this thesis the focus will be on food aid. Food aid is driven by food insecurity, food insecurity is the condition when the fundamental dimensions of food security are disrupted, namely availability, access, utilization, and stability [5]. Food security can be classified into chronic and acute, these concepts, while overlapping have some key distinct characteristics. Chronic food insecurity is when food insecurity persists over time. This type is largely driven by structural causes such as poverty, marginalization, and inadequate access to basic services [6]. Leading causes of food insecurity are inflation, climate change and regional conflict [5]. Food security can be measured using the integrated food security phase classification index (IPC), which is a scale of 1 (minimal/none) to 5 (famine). The food demand within a HO seeks to match this insecurity, the most common bottleneck is funding and therefore is one of the key drivers of demand [7, p. 6]. Food assistance is delivered through two primary modalities: in-kind food distribution and cash-based transfers [2], [28]. The choice is highly context-dependent; cash-based transfers are used when a local economy is developed enough to allow beneficiaries to purchase food [5]. When local markets are not functional or food is unavailable, in-kind food commodities are distributed directly. While cash-based transfers are a significant portion of aid, for the WFP, in-kind food distribution currently constitutes a slight majority of its assistance portfolio [5]. Most of the funding for large HOs comes from contributions from governments [14]. These are usually monetary, however sometimes direct food donations are given, these are called in-kind contributions [7, p. 138].

With a considerable decline in funding, food aid response has ever more focused on meeting beneficiaries urgent nutrition and food needs, this is in contrast to earlier when larger amount was focused on development work and education [28]. Currently about 70% of funds go to better peoples urgent food needs. A large part of this goes to crisis response with about 75% of funding, with 22% to resilience building and 3% going to trying to solve root causes. In 2024 conflict was the main driver of food insecurity, with approximately 65% of acute food security in fragile or conflict-affected situations [29]. Other drivers of food security are restrictions to humanitarian access, driven by conflict and political inaction [29]. Economic factors is another major driver of food insecurity, with records set in global public debt, increases in interest rates and high food inflation rates [29]. Most of these crises are slow-onset and can be prepared for by building resilience and prepositioning aid, the largest constraint currently is funding [5]. The needs are generally well known, at least the current levels with advancements in needs forecasts [16], however funding can be volatile and therefore drive the volatility of funding constrained demand [5].

2.3 Forecasting in Humanitarian Logistics

Laan, Dalen, Rohrmoser, *et al.* [30], did a study on the accuracy of long term demand planning, analyzing Operational Center Amsterdam of Medecins Sans Frontieres (MSF-OCA) long term aid project planning. This is a relevant study to draw parallels with food demand since both medicine and food are perishable items and therefore

require careful planning. The forecasts produced by MSF-OCA are made on a project level and based on yearly average consumption and current consumption levels. Based on these assessments, future plans are made. Laan, Dalen, Rohrmoser, *et al.* [30] analyzed 19 projects and found that there is a significant positive bias in the forecasts. This makes sense since some level of overstocking is needed for risk mitigation in case of a sudden spike in demand. However, this also comes at a higher cost since medical supplies are perishable items. No study was found that proposed a forecasting model for long-term humanitarian demand, which underscores the importance of this thesis.

The literature on immediate and short-term demand forecasting is richer and more in-depth. Herteux, Raeth, Martini, *et al.* [16] showed that food insecurity can be computed with a large dataset of exogenous variables, utilizing reservoir computing. Data points, such as Ramadan and climate-related indices showed promise in producing accurate forecasts for a horizon of 60 days. Food insecurity is a measure of how much a population is in need of food assistance. Food insecurity significantly influences humanitarian needs, making the results of this study relevant to this thesis. Nonetheless, the forecast horizons used is considerably smaller than the one in this thesis. The study highlights the importance of using exogenous data in complex forecasting tasks within the humanitarian sector. They also showed that their model could be expanded to quantile forecasting.

Fuqua and Hespeler [17] proposed a novel approach to forecast fuel demand in specific humanitarian crises. Pairing robust principal component analysis with a LSTM model, they were able to outperform traditional statistical methods by a significant margin when evaluating with root mean squared error (RMSE). They also showed the importance of using exogenous variables for accurate predictions, using 8 exogenous datasets. However, they only validated the accuracy for one- and two-step forward accuracies. Moreover, this study focuses on creating forecasts with sparse datasets and rapid training, which are important in the immediate aftermath of a disaster, which is not applicable to our study.

Giedelmann-L, Guerrero, and Solano-Charris [11] proposed a method of modeling food inventory planning with a system dynamics approach. The focus was on the immediate aftermath of a crisis using a case study based on the 2017 Mocoa landslide in Colombia. They showed that the dynamics post-disaster are extremely volatile and dependent on the affected population, environment, and actors involved. The key takeaways were more on supply chain management than relevant factors for a forecast model.

During the writing of this thesis, Ouchtar, t'Serstevens, and Rahman [27] published a report on time-series forecasting for demand planning within the humanitarian sector, forecasting medical items. The authors focused on an undisclosed HO that provides medical, food, and other services. The analysis included pre-processing of data, which included outlier removal and grouping items to remove volatility. The forecasting methods used were statistical and included different versions of moving average, exponential smoothing, ARIMA, and Croston's method. They concluded that simple statistical methods deliver good results as opposed to more complex methods, and the importance of being able to quickly iterate models, which gives an advantage to simpler models. Moreover, appropriate aggregations of items usually yield better

results as opposed to predicting individual items. Seasonality and trend was also analyzed, finding seasonality of 1, 2 and 4 quarters in the timeseries. Most of the demand patterns where erratic without any clear patterns. Other items, such as food commodities where also analyzed, however were not predicted since most of them where in-kind donations, therefore having a smaller need to do demand planning on their side.

Ultimately, demand forecasting matches supply with demand, lowering delivery times, as well as inventory and procurement costs [1]. This is analogous to the for-profit world where demand forecasting plays a similar role; therefore, similar methods can be used in the humanitarian sector. It is, however, worth keeping in mind that the humanitarian food-relief sector is bounded by funding-constraints, poor transport links, and extremely volatile demand. This makes it generally more difficult to forecast demand in the humanitarian sector than in the for-profit world [1].

2.4 Demand Forecasting in the For-Profit Sector

Similarly to the humanitarian sector, in the for-profit sector demand forecasts are needed for efficient supply chain management [1]. Using historical sales data to produce demand forecasts is a common way to plan supply and optimize inventory based on these forecasts.

Statistical and ML models have long been used in demand forecasting, this is usually combined with expert judgment to produce the final supply plans [22]. Baecke, De Baets, and Vanderheyden [22] showed that leveraging these two ways in a quantitative method yielded accuracy improvements across the board. Two-methods, were employed where one method integrated expert judgment from managers as a feature into the models. The other method used the judgmental data as a restrictive post-processing step to the statistical forecasts. The integrated model generally provided improved results across the board. Older studies have generally used statistical models such as ARIMA or exponential smoothing models, however with larger computational performance, more complex ML methods have gained promise [15].

Overall within forecasting literature the Makridakis- or M-competitions [18] have provided good benchmarks for the state of the art forecasting models. The M-competition is a forecasting competition that has been hosted since 1982 [18]. There are other forecasting competitions, but the M-competitions are among the most established with the largest prize pools. Consequently, they are frequently regarded as the benchmark of forecasting competitions.

The most recent M5 competition [18] is a relevant study for this thesis since the proposed problem was a hierarchical demand problem similar to this thesis. Earlier competitions have seen the dominance of statistical models, however, M5 saw the clear advantage of machine-learning based methods. Notably, the top 5 teams all used variations of LightGBM models combining them in different ways, also known as ensembles. The top performer of the M5 competition used an ensemble of LightGBM models, trained on different levels of the hierarchy. In addition, the forecasts were reconciled afterwards; however, the reconciliation method was not mentioned. Hierarchical time series reconciliation [31, ch. 11] is a method of

producing coherent forecasts from base forecasts. This has shown promise in many fields where hierarchical or grouped forecasts are common [32]. Cross-learning was another key takeaway from the M5 competition [18]. Cross-learning is the ability for one model to learn dependencies between multiple time series and is a key advantage of ML methods as opposed to statistical models [15]. Other notable takeaways were the strength of ensemble models, where multiple models were combined to produce more accurate forecasts, sometimes simply using average predictions between multiple models.

Given the need for probabilistic forecasting, another M5-competition was held, specifically for probabilistic forecasts [24]. Here again, the top contributions used LightGBM models. Different methods such as directly predicting the quantiles were used. However, some methods also derived uncertainty intervals with point-prediction residuals from the in-sample set.

3 Methodology

This section will give an overview of the methodology and theory used in this thesis. Firstly, the research methodology will be covered, after which time series forecasting theory and models will be introduced.

3.1 Research Methodology

The primary research methodology chosen for this thesis is design science research methodology (DSRM), specifically for information systems research [25]. This thesis falls well within the square of information science, since we are using a data-driven approach to solve business problems for a HO. Moreover, the solution of building a model to solve this problem is a design science approach. More formally, "Design science creates and evaluates IT artifacts intended to solve identified organizational problems" [25]. IT artifacts encompass methods, models, instantiations, and constructs. In this thesis, the produced artifacts will be forecast models. DSRM has been widely cited and is a well-established research methodology used in the field of information science. As defined in DSRM [25], this thesis will be split into six activities as outlined in Figure 2.

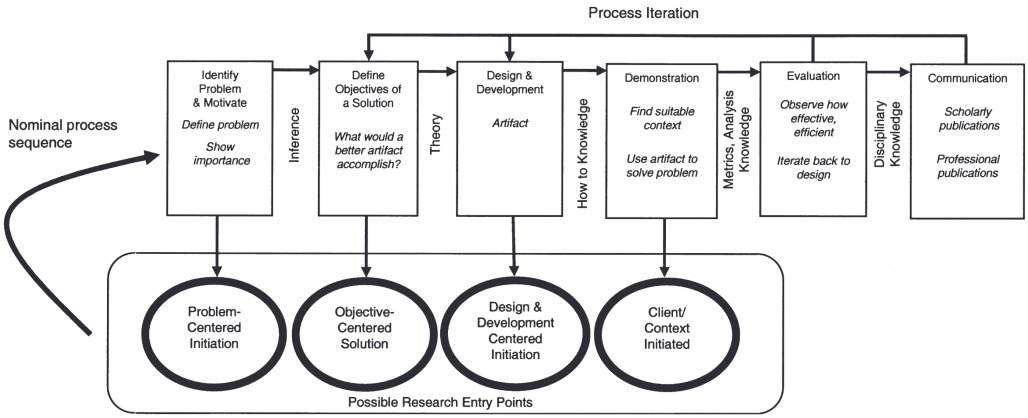


Figure 2: DSRM processes model [25].

These activities are defined in an "nominal process sequence", this means that this is a logical sequence structure for DSRM. However, Peffers, Tuunanen, Rothenberger, *et al.* [25] note that it does not have to be strictly followed and there are multiple entry points. Research entry points are where a study can start, for instance, if a clear problem has been identified, one can directly define objectives for the research. The authors have defined four different entry points. These entry points are based on the context of the research. This thesis will cover all activities and is therefore a "problem-centered initiation".

The first activity, Identify Problem & Motivate, defines a research problem. This serves as the starting point for the creation of artifacts which will serve as the solution. Moreover, the identification and motivation should motivate the reader and researcher

to pursue a solution. This activity should also give relevant context in the field which enforces the need for a solution. In this thesis, the problem of long-term food demand forecasting is introduced in Section 1; moreover relevant context is given in Section 2. RQ1 also serves to give more motivation to the problem and gives a baseline for the existing solutions. The second activity is, Define Objectives of a Solution. The objectives are what the artifact should accomplish; this should be tied to the organizational problem we would like to solve, defined in the first activity. In the case of this thesis, our objective is a higher accuracy for forecasts compared to our benchmark. The objectives are defined in Section 1. Choosing appropriate evaluation metrics is also part of this activity, the evaluation metrics will be defined in Section 3.

The third activity, Design & Development, is the development of the artifact used to solve the defined problem. This is arguably the core of this work and encompasses RQ2. In sections 3, we introduce the theory used to build artifacts, i.e. GBDT forecasting models. The theory will be used for solving the problem of demand forecasting within a HO. The application of theory will be introduced in Section 4. Demonstration is the fourth activity and should show the usability of the artifact. For this thesis, we will demonstrate the solution as WFP case study in Section 4. This is also a part of RQ2. In the fifth activity, Evaluation, we compare the performance of the artifacts; this can be done in many ways but usually includes some quantitative metrics. For this thesis, models are compared to each other with relevant accuracy metrics. Moreover, the robustness and explainability of the model will also be tested, seeking to answer RQ3. Finally, Communication is the last and sixth activity; it compromises how the findings are shared and communicated with relevant parties. The entire thesis serves as the main medium of communication and is therefore also structured as per DSRM activities. Moreover, the findings and implementation are also shared as code to the HO; however, since this is classified information, the specifics will not be shared publicly.

As stated earlier, DSRM is not strictly a linear structure but more often than not iterates through steps based on findings. For instance, in this thesis when developing models, activities 3-5 are repeated based on evaluation findings. Through this robust and well-defined models can be presented based on some level of "trial and error". The iterative steps will be discussed in Section 4. The study will be done based on relevant timeseries and ML literature presented in the following section.

3.2 Time Series Forecasting

The idea that past and present values can be made to predict the future is the basis of time series forecasting. In practice, this means finding and identifying patterns in data that can be used to predict future values. In this thesis, the notation by Petropoulos, Apiletti, Assimakopoulos, *et al.* [15] is used, time series will be notated as a sequence of values $Y = \{y_t : t \in 1, \ldots, T\}$, where T is the number of occurrences. Based on these values, we want to predict future values in time $T + h$, where h is the number of horizons in the future we want to predict. Predictions are notated as $\hat{y}_{T+h}$. Predictive methods are referred to as models f, for instance, a simple moving average (SMA)

model predicts future values by,

$$\hat{y}_{T+1} = f_{SMA}(Y, k) = \frac{1}{k} \sum_{t=T-k+1}^{T} y_t,$$

where k is the number of time steps from which the mean is calculated, k is known as a parameter.

Multi-step ahead forecasting reefers to the practice of forecasting more than one horizon i.e. $h > 1$ [15]. Generally, this can be done by directly predicting a certain horizon $t \in [T+1, T+h]$, known as direct forecasting, or by recursively predicting each horizon, known as recursive forecasting.

3.2.1 Direct Forecasting

A direct forecasting approach predicts multiple steps in the future by fitting one model to each forecasting horizon $t \in [T+1, T+h]$, this means a direct forecasting model will consist of h different models.

$$\begin{aligned} \hat{y}_{T+1} &= f_{T+1}(Y) \\ &\vdots \\ \hat{y}_{T+h} &= f_{T+h}(Y). \end{aligned} \tag{1}$$

3.2.2 Recursive Forecasting

Recursive forecasting uses a general model to predict one step in the future based on previous predictions.

$$\begin{aligned} \hat{y}_{T+1} &= f(\{1, \ldots, y_T\}) \\ \hat{y}_{T+2} &= f(\{1, \ldots, y_T, \hat{y}_{T+1}\}) \\ &\vdots \\ \hat{y}_{T+h} &= f(\{1, \ldots, y_T, \hat{y}_{T+1}, \ldots, \hat{y}_{T+h}\}). \end{aligned} \tag{2}$$

3.2.3 Time Series Characteristics

A time series can exhibit many characteristics, these are inherent properties of the time series which influence its behavior. The simplest models decompose a time series into three components: trend, seasonal and residual [15]. The trend is the smooth underlying change in mean of the timeseries, trends can be split into non-linear and linear trends. For instance if a humanitarian crisis is coming to an end the demand has a negative trend because aid supplies are needed less, where as a humanitarian scale-up might have a exponential positive trend. A time series where a pattern is repeated every "season" has a seasonal component. The season can have varied lengths depending on the data, for instance food prices can have a annual seasonal component, where they are cheaper at harvest [13]. The residual is the inherent randomness of

the time series and can be thought of as noise, this is very hard if not impossible to predict [15].

A important concept within time series forecasting is stationarity. A time series is stationary if it is homoscedastic and does not have a trend (constant mean). Homoscedasticity implies a constant variance, the opposite being heteroscedastic. Stationarity is important when identifying whether or not a time series exhibits trends over a longer period of time; moreover, some models require the assumption of stationarity. Non-stationary time series contain a unit root, which exists when $|\alpha| = 1$ in

$$y_t = \alpha y_{t-1} + \epsilon.$$

A commonly used test for stationarity is the Augmented Dickey-Fuller (ADF) [33] test, which tests the existence of a unit root. The null hypothesis being the $|\alpha| = 1$ and alternative hypothesis being $|\alpha| \neq 1$.

A common way to characterize demand timeseries is to use a four-way classification method based on average demand interval (ADI) and coefficient of variation (CV^2). Where ADI is the average time between demand nonzero values and CV^2 is the variation divided by the mean squared of a timeseries. Introduced by Syntetos, *et al.* [34], this method categorizes timeseries into smooth, intermittent, lumpy and erratic timeseries based on ADI and CV^2 where the thresholds are visualized in Figure 3. This classification method was originally proposed to choose appropriate statistical

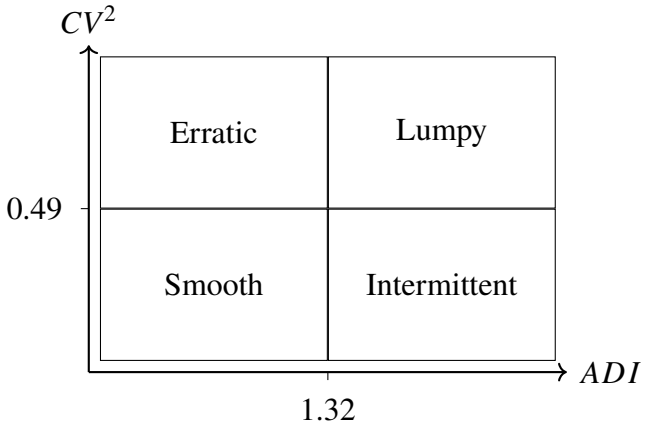


Figure 3: Decision regions for demand pattern classification.

demand forecasting models. For this thesis, the same classification is employed as a description of the data. Since this classification model is widely used, it will serve as a comparison between other demand datasets, such as the ones used in Ouchtar, t'Serstevens, and Rahman [27].

3.2.4 Exogenous Variables

Forecasting is dependent on past data of the variable predicted; however, accuracy can be increased by adding other data points, called exogenous variables [15]. These data points are not being predicted but are added on the assumption that they

correlate with the variable being predicted and thus increase the model accuracy. For instance, Herteux, Raeth, Martini, *et al.* [16] showed that adding weather data and macroeconomic features increased the accuracy of the models. Adding exogenous data to models is not trivial and therefore not all models are able to incorporate this type of data. Moreover, usually preprocessing is needed to transform the data into a usable format; this is also known as feature engineering.

Feature engineering is the process of creating new features from existing data. This can include creating lagged values, i.e. using certain past values with a specified lag to predict future values. Features can also be combined to create new interaction terms, for example by calculating ratios or sums. Additionally, features can be altered or created via transformations; a common transformation is to normalize features with different scales. Taking the logarithm is also a way to generate less skewed data. Transforming heteroscedastic to homoscedastic data is an example of feature engineering for statistical methods such as ARIMA [15]. When fitting or training predictive models on timeseries, it is important to keep in mind the inherent time constraint. This means not "leaking" future values into the past during training. For instance, normalizing by the entire dataset can constitute time-leakage; therefore, when normalizing timeseries, it is common to use a rolling normalization factor of past values [15].

3.3 Machine Learning in Timeseries Forecasting

Machine learning can be defined as a framework for estimating functions on a training sample. In this thesis, the definition of Friedman [35] is used. Consider the response variable $Y = \{y_1, \dots y_T\} \in \mathbb{R}^T$ and explanatory variables $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\} \in \mathbb{R}^{p \times T}$, the training set is defined as $\{y_t, \mathbf{x}_t\}_1^T$ of known values. The goal is to create an approximation $\hat{f}(\mathbf{x})$ of the function $f^*(\mathbf{x})$ mapping $\mathbf{x}$ to y. This is done by minimizing a pre-defined loss function $L(y, f(\mathbf{x}))$ over all $(y, \mathbf{x})$. This can be formalized as

$$f^* = \arg\min_f \mathbb{E}\, L(y, f(\mathbf{x})). \tag{3}$$

Since this problem assumes known response variables also referred to as "ground truth", this constitutes a supervised problem. Supervised methods are models where we can directly compare predictions with response variables with respect to a loss function L, as opposed to unsupervised methods where the response variable does not exist [36, p. 29]. Given the availability of historical response variables, most time series prediction problems, including those in this thesis, are treated as supervised learning problems. Furthermore, ML methods can be split into regression and classification tasks, depending on the need and structure of the data. Classification methods seek to assign y to a specific number of predefined classes. Regression methods, in contrast, predict real-valued numbers $\hat{y}$. In this thesis, only regression methods are considered, since we seek to predict numerical demand volumes. Data is generally split into three different subsets, train, validation, and test, when considering a ML problem. Models are trained on the train sets and iterated based on the accuracy of the validation set. To

avoid overfitting the model to a certain dataset, the final accuracy is measured based on the test set [36, ch. 2].

ML methods usually have many parameters that need to be set before training, these are called hyperparameters and can include learning rate, sampling, regularization, complexity of model and more. There are methods of optimizing for hyperparameters by iterating over the validation set accuracy. This process, often called hyperparameter tuning, can increase the model's predictive accuracy. However, there is a risk of overfitting to the validation set; this occurs when the model becomes too tailored to the validation data, potentially leading to a decrease in performance on the test set, even if validation set accuracy remains high. Most ML models have parameters to decrease the risk overfitting, one widely used method is regularization, notably L_1 (lasso) and L_2 (ridge) [36, p. 61-73]. These work by adding a regularization term to the loss function which penalizes feature weights

$$L_1 = Loss + \lambda_{L_1} \sum_{i=1}^{n} |w_i|, \quad L_2 = Loss + \lambda_{L_2} \sum_{i=1}^{n} w_i^2,$$

where λ are the regularization parameters, one can also include both L_1 and L_2 regularization [36, p. 61-73].

ML methods usually require large amounts of training data. Therefore, forecasting small and complex time series will usually result in sub-optimal results. Statistical methods are commonly fitted to individual time series; in contrast, ML methods can be applied in a cross-learning way, where one model is trained on multiple different time series. Cross-learning not only offers a larger training dataset, but can add relationships not seen in some datasets to others, making the model in some cases more accurate [18].

3.3.1 SHAP for Model Explainability

A major drawback with ML methods is their "black-box" nature. Since ML methods are unstructured and highly complex, it is difficult to know "why" a certain output was produced. Moreover, the stability and robustness are more difficult to infer than more traditional methods [23]. A popular method for describing the relationships between data for an ML model is SHAP values. SHAP is a framework for interpreting model predictions by calculating feature importance using Shapley values [37]. Shapley values are based in game theory and tell us how to fairly distribute a payout among players based on a total payout or value. This value is defined by a value function v, and given a subset of players S, the total value or worth of these players can be computed by $v(S)$. The Shapley values then explain how much of this total payout should be given to each player. In the case of machine learning, the value function is the model and players are features. Note $v(\emptyset) = 0$. The Shapley value for feature j, given a value function v is

$$\phi_j(v) = \sum_{S \subseteq \{1,\dots,p\} \setminus \{j\}} \frac{|S|!(p - |S| - 1)!}{p!} (v(S \cup \{j\}) - v(S)),$$

where S is the subset of players, i.e. features used in our model. Shapley [37] defines a fair payout when it satisfies the properties of Efficiency, Symmetry, Dummy, and Additivity. Let N be the set of all players.

Efficiency: The sum of all Shapley values of players equals the value function of the total coalition $\sum_{i \in N} \phi_j(v) = v(N)$.

Symmetry: Given two players j and i, where $v(S \cup \{i\}) = v(S \cup \{j\})$ for all $j \in N$ and $i \in N$ where $i \notin S$ and $j \notin S$, then $\phi_j(v) = \phi_i(v)$.

Dummy: If a player j does not change the value function, regardless of which coalition it is added to, then $\phi_j(v) = 0$.

Additivity: For a game with combined value functions v^+ and v, the Shapley values are $\phi_j + \phi_j^+, \quad \forall j \in N$.

A major downside is the computational cost of Shapley values, more precisely the computational complexity grows by $O(2^n)$. Therefore, the values have to almost always be approximated [38, ch. 18]. A common tool for calculating these approximations is the SHAP framework [23]. This is called a local model agnostic method, meaning it can be applied to any model for specific predictions. However a global overview can be visualized by plotting all local Shapley values for instance in a beeswarm plot [38, ch. 18].

3.4 Statistical Models

In this thesis, models are categorized into naive, statistical and ML methods. Lines between these are not trivial to draw and the convention of [15] will be used. Naive methods are models that are very simple and range from simply predicting future values as the last seen values to moving averages. Statistical methods use predictive models based on sets of predefined mathematical formulations, examples of this are ARIMA and exponential smoothing models. Finally, ML models are defined as models that do not generate data based on a set of equations, thus allowing for the automatic learning of relationships between data. In this thesis, GBDT models are categorized as ML models.

3.4.1 Exponential Smoothing Models

This subsection introduces the exponential smoothing method, introduced by P. R. Winters and C. C. Holt [39], [40]. Building upon SMA, they introduced a model which predicts future values as weighted averages of past values, by decreasing each past value exponentially via parameter $\alpha \in [0, 1]$

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha) y_{t-1}. \tag{4}$$

If we want to predict for multiple horizons in the future, the future horizon h can be predicted iteratively by

$$\hat{y}_{t+h} = \alpha \hat{y}_{t+h-1} + (1-\alpha)\hat{y}_{t+h-2}. \tag{5}$$

Equation (5) can be further expanded by adding a seasonal and trend component; these can be added communicatively or additively. In this thesis the additive model is used. This is done by decomposing the model into a smoothing, trend and seasonal component known as Holt-Winters' additive exponential smoothing (ES) model. This can be formulated as

$$\hat{y}_{t+1} = \ell_t + b_t + s_{t+1-m},$$

$$\begin{aligned}
\ell_t &= \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\
s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}
\end{aligned}$$

where b_t is the estimate of the trend (slope) of the time series, $\beta \in [0,1]$ is the smoothing parameter of the trend, s_t is the seasonal component with a smoothing parameter $\gamma \in [0,1]$. m refers to the seasonal time between each seasonal pattern. Similarly, as in Equation (5), ES can forecast multiple time horizons iteratively.

The parameters of all exponential smoothing methods can be fit by minimizing the sum of squared errors or SSE. Where

$$\text{SSE} = \sum_{t=1}^{T}(y_t - \hat{y}_t)^2 = \sum_{t=1}^{T}\epsilon_t^2, \tag{6}$$

this is a non-linear minimization problem that can be solved via different optimization algorithms.

3.4.2 ARIMA

Auto regressive integrated moving average (ARIMA) introduced by Box, Jenkins, Reinsel, *et al.* [19] is a popular forecasting model using a combination of autoregressive and moving average models to predict future values. ARIMA is a generalization of the ARMA model which seeks to predict stationary time series. The autoregressive model can be formulated as

$$y_t = c + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t, \tag{7}$$

where p is the order i.e. how many values in the past we will include in the model, ϵ_t is the white noise at time t, $\phi_1, \ldots, \phi_p$ being the fitted parameters and c being a constant (mean) of the time series.

The moving average model predicts future values by taking the weighted sum of past errors, formulated as

$$y_t = c + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}, \tag{8}$$

where q is the order i.e. how many errors in the past we will include in the model, ϵ_t is the white noise at time t, $\theta_1, \ldots, \theta_q$ being the fitted parameters and c being a constant (mean) of the time series.

Summing Equation (7) and Equation (8) yields the ARMA model defined as

$$y_t = c + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \epsilon_t, \quad (9)$$

which can be fitted to a stationary time series by setting the parameters p and q. The ARMA model can furthermore be generalized to non-stationary time series by first creating a stationary time series, predicting the values and inverting the differencing by what is here called integration. Differencing removes trends by taking the difference of consecutive values, written as

$$y'_t = y_t - y_{t-1} = y_t - By_t = (1 - B)y_t,$$

where B is defined as a backward shift operator, introduced for easier notation. Creating a stationary time series can require consecutive differencing, depending on the data. Multiple differencing can be expressed as

$$\begin{aligned} y''_t &= y'_t - y'_{t-1} \\ &= (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \\ &= y_t - 2y_{t-1} + y_{t-2} \\ &= (1 - B - B^2)y_t \\ &= (1 - B)^2 y_t. \end{aligned}$$

In general, differencing of order d can be expressed as $(1 - B)^d y_t$.

The trend of a time series can be removed via differencing, but the data may still be heteroscedastic. Heteroskedasticity can be removed by taking the logarithm of the time series. In conclusion, a time series can generally be made stationary by differencing and log operations. Combining differencing and Equation (9) yields the ARIMA model defined as

$$y'_t = c + \theta_1\epsilon_{t-1} + \theta_2\epsilon_{t-2} + \cdots + \theta_q\epsilon_{t-q} + \phi_1 y'_{t-1} + \phi_2 y'_{t-2} + \cdots + \phi_p y'_{t-p} + \epsilon_t. \quad (10)$$

Equation (10) is formulated by lag operators as

$$(1 - \phi_1 B - \cdots - \phi_p B^p)(1 - B)^d y_t = c + (1 + \theta_1 B + \cdots + \theta_q B^q)\epsilon_t,$$

when fitting ARIMA models the convention of ARIMA(p, d, q) is used. Auto-ARIMA is a algorithm which automatically fits p, d, q based on the timeseries. It first identifies the appropriate order of differencing d using unit-root tests. Where after different p and d are iterated through choosing the most appropriate ones based on Akaikes's Information Criterion [41].

3.5 Decision Tree Models

GBDT models seek to minimize the loss function by partitioning the features space $\mathbf{x}$ into J disjoint sets $R_1, \ldots, R_J$ such that y_i is predicted as based on the region R_j to which $\mathbf{x}_i$ belongs [36, p. 305-308]. In the context of a regression tree, this yields a piecewise linear equation $\hat{f}$. This can be defined by a linear combination:

$$f(\mathbf{x}) = \sum_{j=1}^{J} b_j 1(\mathbf{x} \in R_j), \tag{11}$$

where $1(\mathbf{x} \in R_j) = \begin{cases} 0 & \mathbf{x} \notin R_j \\ 1 & \mathbf{x} \in R_j \end{cases}$ is the indicator function determining if $\mathbf{x}$ is within a given set. $\{b_j\}_1^J$ are the weights, and $\{R_j\}_1^J$ represent the constant prediction value in each respective region $\mathbf{x}$. The set of regions $\{R_j\}_1^J$ and coefficients $\{b_j\}_1^J$ constitute the parameters fitted from the training data. The regions R_j are formed by recursive partitioning of the input space $\mathbf{x}$. Initially, the entire input space represents a single region, this is also called the root node of the tree. From this, new child regions R_j are created by optimally splitting the earlier parent region. These splits are optimally chosen by minimizing a predefined loss function over the region R_j. The splitting is continued until a certain stopping criterion is met, the last regions are called leaf nodes. For instance, the parameters max_depth and num_leaves are two such stopping conditions. max_depth sets the maximum number of splits from the root node to a leaf node. num_leaves specifies the number of end nodes or leaves. Highly complex trees with large depth or number of leaves can lead to overfitting, and conversely shallow trees to underfitting. One way to control the overfitting is to set a min_data_in_leaf parameter. This parameter sets the minimum number of training samples in a leaf, if this minimum set is reached, no further splits are performed [36, p. 305-308].

3.5.1 Gradient Boosting Decision Tree

This section introduces a variant of decision tree models called gradient boosting decision tree (GBDT). This is a multi-tree model consisting of M weak learners which are iteratively calculated based on a loss function. The iterations follow the steepest descent with respect to the gradient of the loss function. GBDT was introduced by Friedman [35], where the problem is defined as a "predictive learning problem".

GBDT estimates f^* formalized in 3 by only considering a set of parameterized functions $f(\mathbf{x}, P)$, where $P = \{P_1, \ldots, P_k\}$. These functions can be used in an "additive expansion"

$$f(x; \{\beta_m, \mathbf{a}_m\}_1^m) = \sum_{m=1}^{M} \beta_m h(\mathbf{x}; \mathbf{a}_m), \tag{12}$$

where h is a general function parametrized by the parameter vector $\mathbf{a} = \{a_1, a_2, \ldots\}$. However, in practice, when using a finite data sample $\{y_t, \mathbf{x}_t\}_1^T$, one cannot accurately

estimate $\mathbb{E}_y[\cdot|\mathbf{x}]$ for each x_i. For this, a method of parametrized optimization is proposed where 12 is expanded to

$$\{\beta_m, \mathbf{a}_m\}_1^M = \arg \min_{\{\beta', \mathbf{a}'\}_1^M} \sum_{i=1}^{N} L(y_i, \sum_{m=1}^{M} \beta' h(\mathbf{x}_i, \mathbf{a}'_m)). \tag{13}$$

Equation (13) defines a complex and large optimization problem; solving this directly can be infeasible, especially with a large number of additive components M and depending on the nature of the loss function L. One possible solution is using a "greedy stagewise" approach

$$(\beta_m, \mathbf{a}_m) = \arg \min_{\beta, \mathbf{a}} = \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}_i) + \beta h(\mathbf{x}_i, \mathbf{a})).$$

where the estimated function can be computed as

$$f_m(x) = f_{m-1}(x) + \beta_m h(x, \mathbf{a}_m).$$

Here f_M is the final additive model, where f_0 is called the "base learner" and each additive step $\beta_m h(x, a_m)$ is called a "boost". One can reach a local optimum by iteratively calculating f_m and minimizing the loss $L(y_i, f_{m-1}(\mathbf{x}_i))$. This can be done by computing the unconstrained gradient defined as

$$-g_m(x_i) = -\frac{\partial L(y_i, f_{m-1}(\mathbf{x}_i)}{\partial f(\mathbf{x}_i)},$$

which gives the steepest descent direction $-g_m$. However, the gradient is only defined at data points $\{x_i\}_1^N$ and therefore is not defined for other x-values. One can solve this by choosing a $h(\mathbf{x}; \mathbf{a}_m)$ that is most parallel to $-g_m$, which generalizes the solution to all $\mathbf{x}$ by

$$(\beta_m, \mathbf{a}_m) = \arg \min_{\mathbf{a}, \beta} \sum_{i=1}^{N} (-g_m(\mathbf{x}_i) - \beta h(\mathbf{x}_i; \mathbf{a}))^2.$$

From this, a generalized steepest descent search can be formulated, also known as the unconstrained line search

$$\rho_m = \arg \min_{\rho} \sum_{i=1}^{N} L(y_i, f_{m-1}(\mathbf{x}_i) + \rho h(\mathbf{x}_i; \mathbf{a}_m)).$$

After which the estimated function can be updated as

$$f_m(x) = f_{m-1}(\mathbf{x}) + \rho_m h(\mathbf{x}; \mathbf{a}_m).$$

For this algorithm, any differentiable loss function can be used. Most commonly standard least squares is used, due to its computational efficiency. h can in principle

be generalized to any parametrization function; in this case, in this case we apply it to regression decision trees as defined in 11, yielding

$$f_m(x) = f_{m-1}(\mathbf{x}) + \rho_m \sum_{j=1}^{J} b_j 1(\mathbf{x} \in R_j), \tag{14}$$

where b_j and R_j are the parameters to be fitted. Solving directly for 14 works well for the training set, but this can quickly lead to overfitting which reduces the generalizability of the model. To counter this, it has been found that regularization is a good approach by

$$f_m(x) = f_{m-1}(\mathbf{x}) + v \cdot \rho_m \sum_{j=1}^{J} b_j 1(\mathbf{x} \in R_j),$$

where v has been added to 14 which can be tuned for $0 \leq v \leq 1$. Where v is commonly referred to as learning_rate, a higher v can lead to overfitting and conversely a smaller v can lead to underfitting. Another parameter that is commonly tuned in GBDT is M which is the number of decision trees in the model, also known as num_estimators.

3.5.2 LightGBM

Proposed by Ke, Meng, Finley, *et al.* [21], LightGBM is a GBDT model that introduces novel features to traditional GBDT to increase efficiency and accuracy. LightGBM expands upon GBDTs by adding two features: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). Ke, Meng, Finley, *et al.* [21] showed that utilizing GOSS and EFB sped the training process by up to a factor of 20 while retaining similar accuracies on multiple public datasets.

Sampling is a commonly used strategy in ML to reduce overfitting and increase training speeds [21], by only using a subset of the training set for an iteration of training (boost in GBDT). However, within GBDT there is no inherent weight given to a feature. To introduce sampling to GBDT Ke, Meng, Finley, *et al.* [21] propose a way of measuring "importance" of a feature based on their corresponding gradients in the training process. Where larger gradients, which will ultimately lead to the largest loss minimization, are kept and features with a smaller gradient are not included. The logic being that features with small gradients are already "learned" or "trained" and high gradient features are "under-learned" [21]. This method of keeping high gradient features and randomly dropping "learned" or low-gradient features is called GOSS.

The other novel approach introduced in LightGBM is EFB. EFB introduces a nearly lossless way of bundling certain features, which effectively reduces the number of features in training. Since there are usually sparse features that are seldom non-zero at the same time, these features can be combined into one. This can greatly speed up training times without loss of information.

3.5.3 Loss Functions

Different loss functions can be used, such as squared loss, or simply absolute error [42]. Loss functions have a large influence on the model training since they define

the gradient by which the model will be optimized. Therefore, careful selection of loss functions should be considered [42]. Squared losses optimize for the mean and absolute error loss for the median. For squared losses, large values have a bigger impact than losses based on the absolute error; therefore squared losses are generally more influenced by outliers. Losses such as Huber loss or log-cosh have a tradeoff where for small values a squared loss is applied and larger values a linear absolute error-like value is taken [42]. For instance, the Huber loss is defined as

$$L_\delta = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{if } |(y - \hat{y})| < \delta \\ \delta\left(|(y - \hat{y})| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases}, \tag{15}$$

where a δ is defined beforehand. The loss functions should be similar to the evaluation metrics of the model, however this alignment is not always possible. For instance, for some GBDT models such as LightGBM the hessian and gradient have to be calculated, therefore a loss function which does not have a easily calculated gradient or hessian is suboptimal.

3.6 Evaluation and Benchmarking

When developing forecasting models or methodologies, there is usually a need to measure its success. One should be vigilant when benchmarking models since it is common to choose metrics that exacerbate the proposed model's performance. This thesis follows the a benchmarking methodology proposed by Petropoulos, Apiletti, Assimakopoulos, *et al.* [15] which is defined by the following five steps:

1. New methods should be compared to a larger set of suitable benchmark methods.
2. Methods should be compared with a diverse set of metrics.
3. Testing should take into account if differences are statistically significant.
4. Methods should be tested with a rolling sample window.
5. It is advised that the code be produced with a open-sourced programming language such as R or Python.

3.6.1 Point Forecasting Evaluation

In this thesis, we define quantitative measures of success evaluation metrics. One model can have and arguably should have multiple evaluation metrics to get a good overview of the behavior of the forecasts. There are a variety of evaluation metrics for different purposes. The most simple metric is the residual of the model, this is simply the difference between the predicted and true value for a specific timestep t

$$\epsilon_t = \hat{y}_t - y_t. \tag{16}$$

Summing this yields the total of the residuals of a time series, this is sometimes referred to as the bias,

$$\text{BIAS} = \frac{1}{h}\sum_{t=T}^{T+h} \epsilon_t.$$

The bias provides an indication of whether the model tends to underpredict or overpredict, but it does not quantify accuracy because positive and negative errors can offset one another. The mean absolute error (MAE) addresses this issue

$$\text{MAE} = \frac{1}{h}\sum_{t=T}^{T+h} |\epsilon_t|. \tag{17}$$

MAE is a widely used metric for model accuracy, one caveat is that it optimizes for the median [31, sec. 5.8]. In many cases, it might be preferable to optimize for the mean. For instance for intermittent demand data containing a large amount of zeroes, the median would be very low, however one might want to keep a "mean" stock and therefore predict higher values, corresponding to the mean. Optimizing for the mean can be done by evaluating forecasts with squared errors such root mean squared error (RMSE) [31, sec. 5.8]

$$\text{RMSE} = \sqrt{\frac{1}{h}\sum_{t=T}^{T+h} \epsilon_t^2}.$$

Metrics like MAE or RMSE give a good sense of accuracy for the model, however, when comparing multiple time series with different scales a relative or scaled error is commonly used. This is so that larger timeseries do not dominate the error metric, given that we want to measure the relative accuracy of all timeseries equally. A common approach has been dividing the residual by the actual value yielding a percentage-based error. Percentage-based errors work for time series with small nonzero values, but this metric fails when the actuals are zero. To solve this Hyndman and Koehler [43] proposed a scaled error which scales a given metric by a naive model error, scaling RMSE would yield RMSSE,

$$\text{RMSSE} = \frac{RMSE}{RMSE^{naive}} = \frac{\sqrt{\frac{1}{h}\sum_{t=T}^{T+h} \epsilon_t^2}}{\sqrt{\frac{1}{T}\sum_{t=2}^{T}(y_t - y_{t-1})^2}}.$$

Scaled errors have been widely used, especially in intermittent time series with many zeros [18], [43]. Scaled errors also take into account the variability of historical data, by weighing stable series less giving them higher losses. This makes sense since constant series should be easier to predict and therefore be penalized more. Additionally, scaled errors exhibit greater stability compared to metrics like relative errors, which may exhibit infinite variance [43].

It is crucial that evaluation metrics align with business requirements, and the metric's explainability is also a key factor, especially for a non-technical audience [15]. Sometimes these criteria align but most often not. For instance, MAE is much more logical than RMSE for most people to understand. For this reason, it is usually needed to use several metrics for different purposes.

3.6.2 Probabilistic Forecasting Evaluation

For evaluating probabilistic forecasting, one cannot simply just look at one point, but should account for the predicted distribution. For quantile predictions, one can calculate the score for each quantile, known as the quantile loss, or pinball loss (PL) [31, sec. 5.9]

$$\mathrm{PL}_\tau(f_{t,\tau}, y) = \begin{cases} \tau(y - f_{t,\tau}), & y \geq f_{t,\tau} \\ (1-\tau)(f_{t,\tau} - y), & y < f_{t,\tau}. \end{cases} \tag{18}$$

Where τ is the quantile and $f_{t,\tau}$ is the predicted level for quantile τ at time t. PL gives the accuracy metric for one quantile, to get a general metric for a predicted distribution over all quantiles one can simply take the mean pinball loss (MPL) over all quantiles [24]

$$\mathrm{MPL}_{f_t,y} = \frac{1}{n} \sum_{\tau \in Q} \mathrm{PL}_\tau(f_{t,\tau}, y), \tag{19}$$

where n is the number of quantiles and Q is the set of all quantiles. Another metric related to quantiles is referred to as coverage. This metric evaluates the proportion of values that fall below a given quantile; values that are near the quantile are considered favorable.

3.6.3 Significance Testing

Simply taking the mean of the metrics are a good way to get a sense of difference between the models. However, this method can sometimes be flawed and these differences could simply be different by chance. A common approach in testing the statistical significance between two timeseries is the Diebold Mariano test [15]. However this can not be applied to grouped timeseries data as is, moreover this test is best used with a long prediction horizon. More generic testing can also be done by simply looking at the differences in error metrics. One such test is the permutation test. The permutation test is a non-parametric test to see if two metrics are derived from the same distribution ie. same model [44]. The test has the following hypothesis.

- H_0: Prediction errors from both models have equal distributions.
- H_1: Prediction errors from both don't models have equal distributions.

The test works as follows, let e_A be the errors from model A and e_B errors from model B.

1. Calculate $d_i = e_{Ai} - e_{Bi}, \forall i = 1, \ldots, n$
2. Calculate the observed mean difference $T_{obs} = \frac{1}{n} \sum_{i=1}^{n} d_i$
3. Repeat m times:
    a) Flip signs of d_i randomly for all $i = 1, \ldots n$
    b) Calculate $T = \frac{1}{n} \sum_{i=1}^{n} d_i$

4. Calculate the significance level $p = \frac{k}{m+1}$, where k is the rank of $|T_{obs}|$ among all $|T|$ ordered from low to high

3.6.4 Evaluation Data

Evaluating a model is an iterative process of trial and error, especially when evaluating ML models which can be optimized with hyperparameter tuning. The best practice is to always keep one part of the data outside of validation and training; this will be referred to as the test set sometimes called the hold-out set [31, sec. 5.8]. For some models, for instance ARIMA models, a sufficient split is train/test data, where the model is fitted on the train set, evaluated and then finally when parameters are chosen, tested on the test set. However, models such as GBDTs usually need another set, the validation set. Where models are trained on the train set, then the model is used to predict the validation set, accuracy is measured, and hyperparameters are tuned accordingly. This is done until a sufficient accuracy is reached. Then the models are finally tested on the test set. This kind of split minimizes the risk of overfitting the model to a specific time.

A common method in ML to reduce overfitting and expand the train/validation set is to use cross-validation [15]. Where the train/test set are permuted and retrained, the average accuracy is then calculated and the model evaluated. Usually, random permutations are sufficient; however, in time series, where there is a temporal order constraint, this is not possible, since the train set always needs to be before the validation or test set. Multiple time series specific cross-validations have been introduced by keeping the temporal order. Expanding-window cross-validation keeps the same training start time but moves the end training period forward for each fold (see Figure 4). This is the most common time series cross-validation strategy; however, other cross-validations can also be used. Cross-validation is crucial to building robust and accurate models and this was one of the key takeaways from the M5 accuracy competition [18], where all top teams tuned their models using an undisclosed cross-validation strategy.

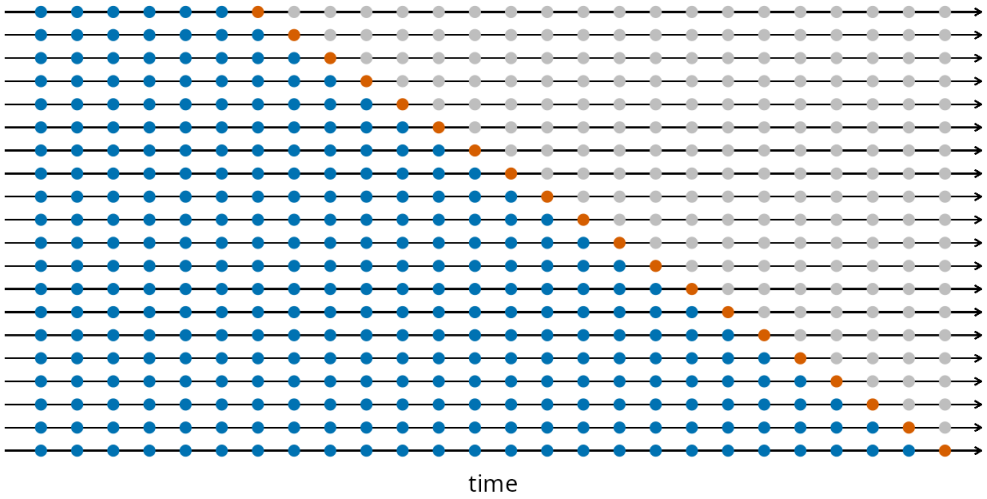


Figure 4: Expanding window cross-validation train/test sets [31, sec. 5.10]. Blue dots are within the train set, orange test set and grey are kept out.

4 Case Study

This thesis presents a case study on forecasting food commodity demand to support and enhance the World Food Programme's (WFP) global supply chain planning. This has been done in collaboration with WFP's Supply Chain Planning & Optimization Unit. To estimate demand, historical data of 'handovers to cooperating partners' will be predicted. Handovers are the amount of particular commodities that WFP delivered to its cooperating partners consisting of local and international non-profit organizations. These cooperating partners are then responsible for the distribution to beneficiaries. This data is categorized by 'activity type', which includes different WFP activity categories like general food assistance, malnutrition prevention, and school feeding, among others.

The data is per country office (CO), commodity, activity type and period. More in depth explanation of the target data is introduced in Section 4.2.1. This section gives an overview of the case and also how the study will be conducted in practice. An overview of the data is given, after which the model selection process is introduced. Then the evaluation process is described, this process has the aim of answering RQ1 and RQ2. After which, the methods analyzing the robustness and explainability of the models is introduced, this is aimed at answering RQ3. Finally, a probabilistic modeling approach will be implemented based on the chosen model to answer RQ4.

4.1 Overview

In WFP, COs can purchase commodities directly from suppliers or prepositioned WFP corporate inventory. Moreover, COs can also receive direct commodity contributions, known as in-kind contributions. These commodities are then handed over to cooperating partners, who then distribute the aid to beneficiaries. Implementations can happen through various channels, including programs such as school feeding, crisis response, resilience building, and others. These are collectively referred to as activity-types. A simplified overview of the WFP distributional network for one country office has been visualized in Figure 5.

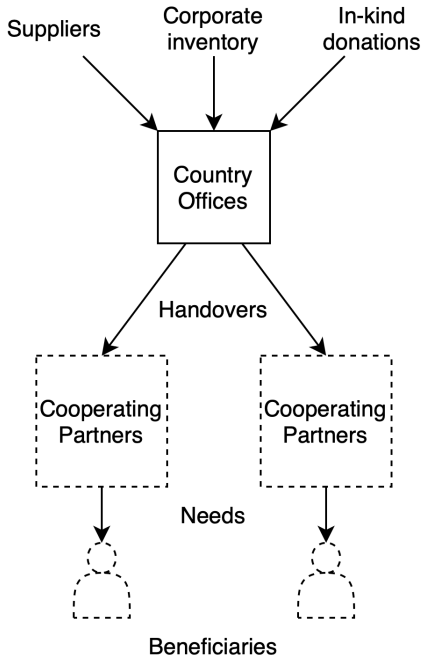


Figure 5: WFP country office simplified supply chain aid distribution network, the actual network is much larger.

COs make plans for how much food they expect to distribute for each month in the future known as implementation plans (IPs). These plans are based on the needs of beneficiaries and take into account funding and operational constraints to reach a realistic IP. No food is assumed to be needed if no plan is submitted. Food commodities can be stored in local inventories, therefore implementations and purchases can have a time lag depending on the local inventory levels and prepositioning strategies by the countries. While following an overarching framework, these plans are generally developed independently by each country office, with variations in structure and approach reflecting the specific contexts and practices of each CO. This is also apparent in the difference in the profiles of implementation plans between countries. Note that these are not used directly as demand forecasts for the purpose of corporate inventory prepositioning. This is due to the IPs having varying accuracy across different countries and are on average positively biased.

Corporate prepositioning can decrease the lead time for COs from when they receive donor funds to the final distribution. This is opposed to procuring from food suppliers after funding is received. It also facilitates cost savings, especially through purchasing in bulk at lower prices and by positioning supplies in corporate inventories, accessible to multiple COs [28]. COs cannot always directly purchase and preposition food due to funding and inventory constraints. However, this can be done at a corporate level in anticipation of CO purchases. Moreover, since funding and needs are uncertain, the future plans and purchases are also uncertain. Therefore, corporate stock prepositioning and upstream supply chain optimization are highly dependent on demand forecasts and their probability distributions. Some demand profiles are assumed to share characteristics between countries. Therefore, a general

forecasting model is proposed, that can "learn" from multiple commodities, countries, and activity types. This creates a grouped multi-horizon forecasting problem.

IPs provide much information about the demand in the future, in particular on planned scale-ups and scale-downs of operations. However, some events leading to scale-ups or scale-downs are unpredictable. Therefore the related changes in demand are not captured in advance in the IPs, nor in any other exogenous data available. Examples of such sudden and unforeseen events can be political shifts, coups, new conflict or natural hazards. The choice has been made to treat these events similarly to other data. In practice, it is not really possible to remove such data, since these events are not clearly defined with a start and end date. Furthermore, scale-ups can restructure the operations, thus permanently changing the nature of the demand data. Ouchtar, t'Serstevens, and Rahman [27] proposed treating events such as scale-ups differently by, for example scenario planning. Scenario planning would be a interesting avenue of future research, however it is out of the scope of this thesis.

4.2 Data

This section introduces the datasets used for model training. The target data is predicted on country, commodity and activity type granularity, these target timeseries are referred to as segments $s \in S$, where S is the set of all segments. The segments have been anonymized via a mapping with prefixes ISO for countries, CO for commodities, and AC for activities. So when a specific segment is referred to, it is of the form (ISO0X, CO0X, AC0X), where X is the anonymization mapping number. The data being used is from January 2016 to March 2025, the periods are monthly and will be shown in month/year format and referred to as Period. The choice of external data points is based on availability and earlier research, notably [3], [16], [17]. The LightGBM model will be trained on historical handover data. To improve accuracy and test the adding of exogenous variables more features will be added. Since there is a rich historical dataset of IPs, these will be included. Moreover, IPC data [6], macroeconomic indicators including GDP growth and inflation indices per country [45]. WFP internal features such as inventory levels and corporate level funding data will also be included.

4.2.1 Handover Data

The target data that we are seeking to forecast is the actualized demand in metric tons (MT), this data is available for the periods 01/2018-04/2025. In Figure 6, the total aggregated demand is shown; this includes all summed segments.

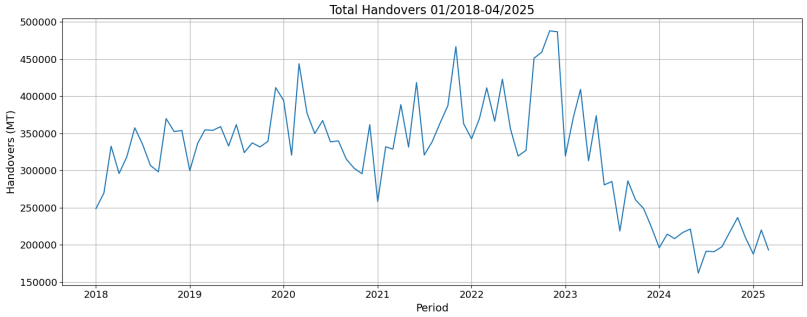


Figure 6: Total demand aggregated over countries, commodities and activities.

Each segment has its own "life-cycle"; this is a distinction between when a segment is active or inactive. Inactive segments refer to the period when a specific commodity in a particular country is not utilized for a given purpose. These pertain to segments without active IPs during that timeframe. Examples of segment life-cycles are shown in Figure 7. The number of unique segments are displayed in Table 1, the amount of unique active segments during the test set (04/2024-03/2025) is also highlighted.

Figure 7: Four different segments, the inactive period is marked with red background.

Table 1: Summary of Unique Values and Segment Combinations

Description	Count
Countries	80
Commodities	47
Activity Types	19
Total Unique Segments	1602
Unique Active Segments During Test Set	817

Looking at total demand data in Figure 6 can be misleading since this aggregate

data is not directly being forecast. Moreover, operations with large volume dominate this plot "hiding" smaller operations. A larger set of segments has been visualized in Section 7. Many of these series are highly non-stationary with different trends, these larger segments rarely have any zero demand values which points to lower demand segments being mostly zero, as can be seen in Figure 7.

Key statistics about the handover data is shown in Table 2. The data has a large number of zeroes with sporadic demand, approximately 50.07% of values being zeroes. Moreover, the dataset is extremely skewed, with most demand being small and some demand being many magnitudes larger. This underlines the volatility of the demand, which is visualized in Figure 8, where the data has been filtered excluding very small values and very large to get a better understanding of the distribution. This reflects the nature of HOs, having many smaller operations and a few large ones, where a lot of assistance is needed. It also poses a major challenge in developing a general predictive model. The distribution of the data is similar to other intermittent demand datasets, such as in the M5 dataset [18].

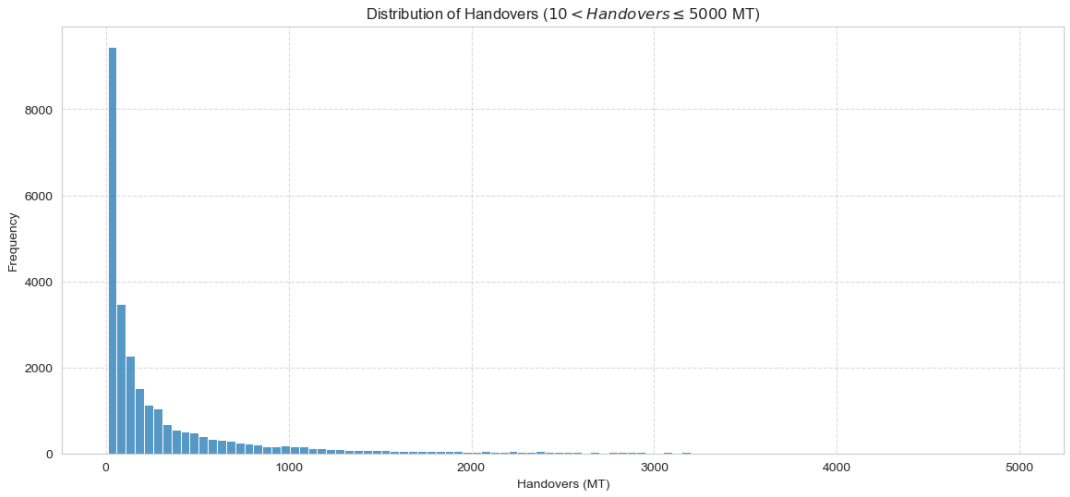


Figure 8: Histogram of actualized demand for demand within $[10, 5000]$MT, 26984 data-points filtered away below 10MT and 1036 above 5000MT.

Table 2: Basic Statistics of Actual Demand

Statistic	Value
Count	75889
Mean	369.93
Std	2884.42
Min	0.00
25% (Q1)	0.00
50% (Median)	0.00
75% (Q3)	54.66
Max	138081.05
Zeroes Count	38477

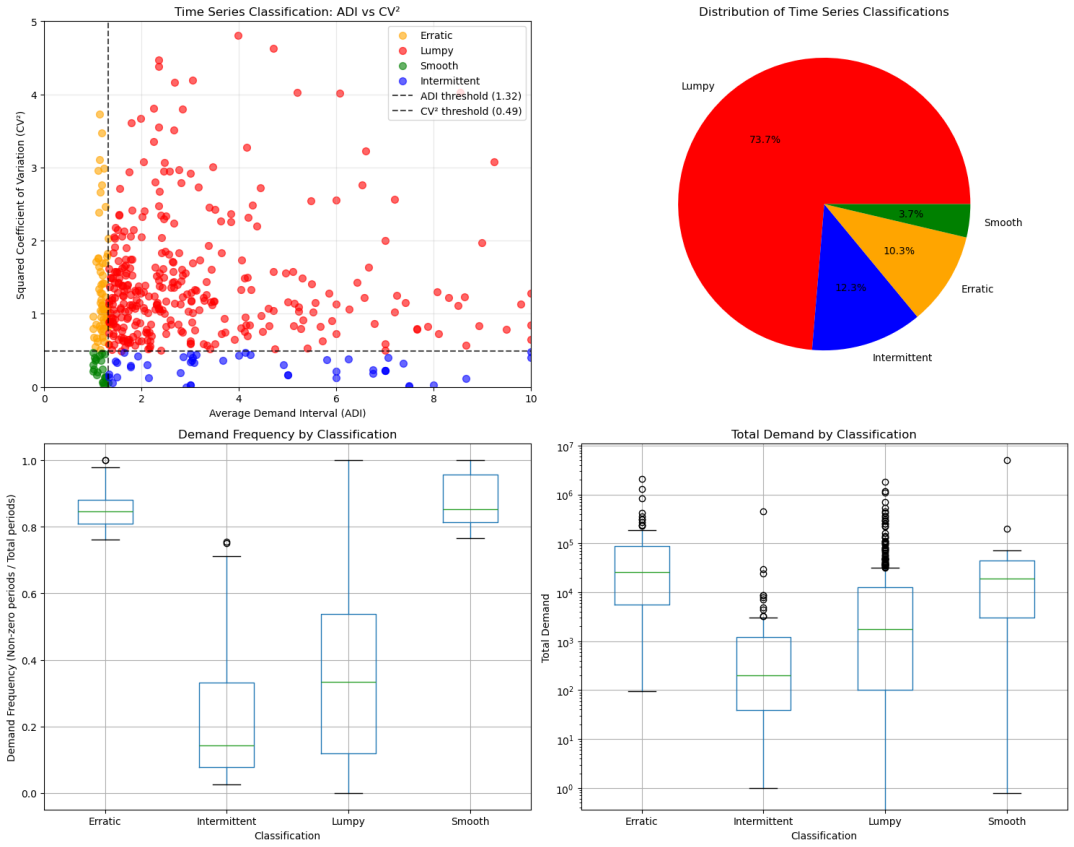


Figure 9: ADI/CV^2 demand patterns classification

To get a more accurate sense of the demand characteristics a widely used 4 class categorization of demand patterns is used [34]. As visualized in 9 around 74% of timeseries are lumpy, 12% intermittent, 10% erratic and 4% smooth. This also highlights the difficulty in forecasting these models especially with statistical models which generally perform quite poorly on other than smooth data.

4.2.2 Implementation Plans

Implementation plans are the demand plans for individual COs. The IPs are derived from a needs assessments combined with funding forecasts and implementation/operational constraints. Examples of IPs have been shown in 10. These plans can give a sense of scale or changes in trends which classical statistical models do not pick up on, therefore, these could provide useful exogenous features. This is analogous to the study by Baecke, De Baets, and Vanderheyden [22], who found that adding judgmental forecasts as a feature, yielded accuracy gains in demand forecasting models.

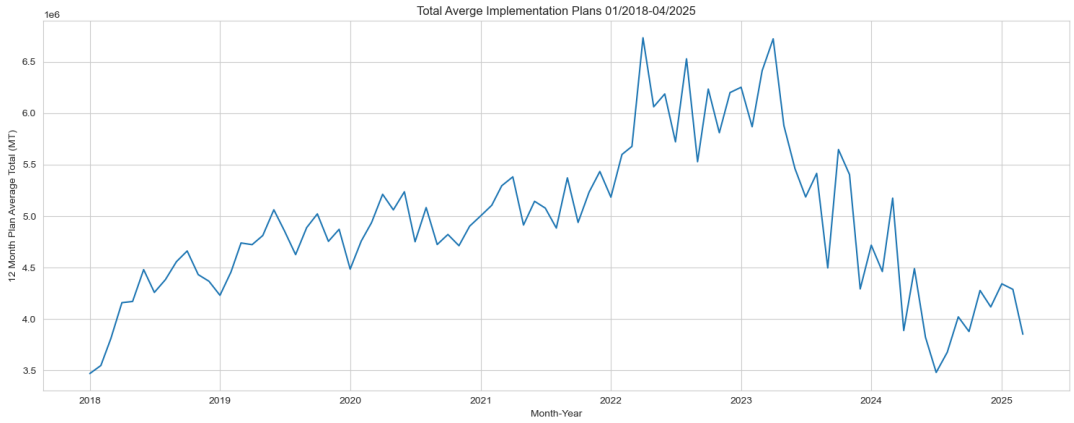


Figure 10: Average over 12 month ad-hoc plans summed.

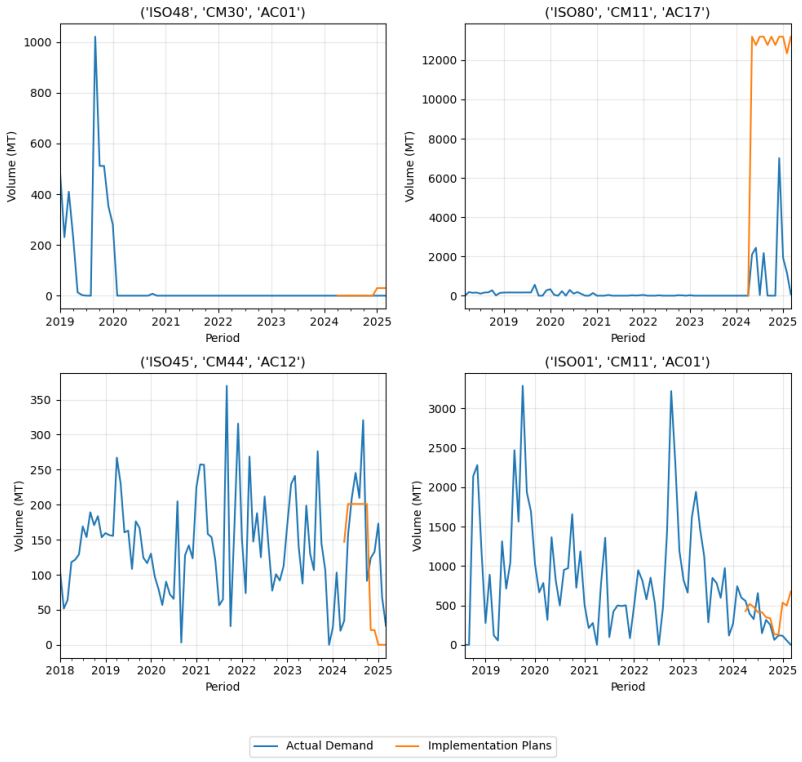


Figure 11: Examples of IPs for the test set.

When analyzing the IPs compared to handovers (see Figures 6 and 11), a consistent positive bias can be identified. This is most likely due to funding needs not being met. This is because many COs create plans based on "best-case" funding scenarios. Therefore, a simple heuristic model is proposed where this bias is minimized by calculating a rolling 'bias ratio' over a specified window N,

$$r_t = \frac{1}{N} \sum_{n=1}^{N} \frac{y_{t-n}}{x_{t,t-n}},$$

where $x_{t,t-i}$ is the IP for time t at reporting period $t - i$. This bias is then removed from the IP for the UIP forecast,

$$\hat{y}_{T+h} = x_{T+h,T} \times r_T. \tag{20}$$

this data will be referred to as the unbiased implementation plan (UIP), and will be compared with the other models. Biases of 3,6,12 months will be compared on the test set and then the best models will finally be tested with backtesting. UIP and the IPs are considered judgmental forecasts, since these are made by individual COs based on the needs and plans of the operations.

4.2.3 Inventory Data

WFP internal inventory data is used for each segment. We consider two main data points: total inventory utilization and the relative utilization based on total inventory.

A country office's inventory level relative to its IP volume gives an indication of its likelihood to implement a higher or lower portion of the plan. For example, if a country has high inventory levels, it is likely they can distribute as much as planned, at least in the first few months, while low inventory makes it more likely they will implement less than the IP.

4.2.4 Annual Organizational Data

WFP global metrics are included as well. These are available on an annual basis for the organization and show large-scale trends in funding, expenditure, and inflation. All data is sourced internally within the organization. All of these metrics have a rough annual forecast that will also be included. These are global funding (USD), global expenditure (USD), global average food prices (USD/MT), food price index, food ratio expenditure, inflation rate, inflation index on other costs, food expenditure (volume index). Note this data includes historical actualized values as well as future forecasts.

4.2.5 Macroeconomic Indicator Data

To test the model's capacity to use external macroeconomic data we have chosen to test inflation and GDP growth from the International Monetary Fund (IMF) World Economic Outlook Report 2025 [45]. This report includes annual real GDP growth data and global inflation rates for average consumer prices. This thesis also includes the outlooks for the year 2025 given in this report [45]. Since food demand is heavily funding-driven [7, p. 36], it is assumed that GDP growth could reflect this; this will become apparent when testing the feature importance in the models. Inflation is included as the cost-of-living crisis is a major contributor to food insecurity globally [5] [5]. These datasets contain historical values as well as future forecasts, data on historical past forecasts where not available. This means that the models are trained on historical values and on inference forecasts will be used.

4.2.6 Food Insecurity Data

Integrated food security phase classification (IPC) [6] data has also been included, this data reflects the number of people within a specific IPC index rating from 0 (food secure) to 5 (famine). This should strongly affect the needs of the people on the ground which is correlated with food demand somewhat [6]. However this data is problematic since it is quite sparse. The IPC data comes from reports published by the Food Security Information Network (FSIN), FSIN publish data on crises stricken areas, therefore data is not available on a majority of countries and periods in the dataset that is being predicted in this thesis. The available rows with IPC data is about 15%, this data is updated infrequently by reporting from food insecure areas in the world. To populate a larger portion of the data the choice has been made to forward fill the rest of the data. This expands the data availability to around 53% of the rows. The IPC values are joined with the reporting period columns, ie. the features will be the latest IPC values available at the time of inference, this way there is no time leakage.

4.3 Model Selection

For fair comparisons, a mix between naive, statistical, judgmental and ML methods should be chosen [15]. For the naive method, three different moving-average models will be initially chosen, with a window of 3,6 and 12 months. For statistical methods Auto-ARIMA [41] and Holt-Winters exponential smoothing (ES) [39] are chosen. These methods are widely used, and are therefore suitable benchmarks. Moreover Ouchtar, t'Serstevens, and Rahman [27], that ES and MA performed relatively well, while ARIMA models performed poorly for humanitarian demand forecasting. Ouchtar, t'Serstevens, and Rahman [27] found that triple and double exponential smoothing where relatively accurate on forecasting humanitarian demand. For a fair comparison, a double, triple and single ES are fit on each timeseries and the best model is chosen by choosing the one with smallest SSE 6. For this study, the analysis was limited to additive ES models. This decision was primarily driven by the nature of the demand data, which is characterized by high intermittency and frequent zero-value periods. Multiplicative seasonality is less appropriate for such series, as seasonal effects are not expected to scale with a level that is often zero or close to zero. IPs and UIP 20 will also be included, and their accuracies compared; these are relevant since they show "judgmental forecasting" models. Choosing this mix of models will give an overview of the performance of naive methods, statistical methods and judgmental methods with the IPs.

The choice of testing ML models for this problem is mainly based on the use of exogenous data and the nature of the timeseries dataset. Since the dataset is large and grouped within different categories, similar to the M5 forecasting competition, there is large potential for cross-learning. Moreover, there is much evidence to support the inclusion of many exogenous variables to improve accuracy [16]–[18]. This strongly points to ML models having the potential to deliver more accurate forecasts as opposed to statistical methods, which have to be fit to individual timeseries and have limited inclusion of exogenous data.

Recently, there have been developments within time series forecasting with ML methods. Methods such as TFT [20] and LSTMs [17] have shown promise and could provide good results. However, these methods most often come with a large computational cost and complex hyperparameter tuning. This leads to slower training times and less possibility to quickly iterate over different versions of the models. As stated by Ouchtar, t'Serstevens, and Rahman [27], models within the humanitarian sector should be quickly iterable. GBDTs have shown large promise for forecasting tasks [18] and are well suited for the type of tabular data used in this case. Moreover, GBDTs, especially LightGBM [21], are very quick to iterate over large datasets. They also make it easy to include exogenous data points. As shown in Makridakis, Spiliotis, and Assimakopoulos [18], LightGBM also leveraged cross-learning very well, which could provide accuracy improvements in our case. Moreover, due to the quick training and inference of LightGBM, these models can be used in ensemble [18], being able to easily be applied in a hierarchical forecasting fashion, predicting timeseries at different aggregation levels and then reconciling [31, ch. 11]. Ensembling also provides the possibility of probabilistic predictions via direct quantile forecasting or parametric

probabilistic forecasting via methods such as LightGBMLSS [46].

The choice of including only one ML method in this thesis is to better finetune and analyze the results. A comparison could be made between multiple different models, however this would "water down" the analysis for this specific case. Since the dataset is complex and requires a large amount of time for feature engineering, the choice of only including different versions of LightGBM has been made. Moreover, since a goal for this case is to move the model into production, the robustness and explainability will be important. Doing this analysis for multiple models would not fit the scope of a master's thesis. For the LightGBM models a direct (multi-model) version and a recursive version will be tested.

4.4 LightGBM Implementation and Feature Engineering

4.4.1 Feature Engineering

LightGBM is the only model that uses exogenous data and therefore requires some feature engineering. A recursive and direct version of LightGBM will be implemented and tested. The recursive model creates one LightGBM instance and fits existing data on this model. During inference, the model will recursively predict each forecast horizon, the past predictions will be fed into the next horizon as lagged values. This is as per Equation (2). The direct model will create h LightGBM instances, each instance will predict one specific future horizon. The instances are analogous to f_i in Equation (1). Most exogenous data are fed into the models as such, for demand, historical demand values lagged and moving average values are created. For the recursive model the lagged handover data is updated iteratively. All the features are shown in Section 7.

4.4.2 Hyperparameter Tuning

For both the direct and recursive models, the same set of hyperparameters will be used. The learning rate v, L_1 and L_2 regularization, number of estimators m, the feature fraction f, number of leaves in the tree J, minimum data per leaf f_J. There are many more hyperparameters that can be chosen for the LightGBM model; however, as a starting point, these are generally considered important for training, especially number of estimators and learning rate. Since the dataset is extremely volatile, regularization terms are added to avoid overfitting. These hyperparameters will be trained first on a validation set with a larger search range, after which they will be tuned via cross-validation over 20 folds in the past. This training seeks to first narrow the scope of the parameter values, after which cross-validation will avoid overfitting. Finally, after cross-validation, the models will be evaluated based on the test set as well as a final cross-validation. The average cross-validation metrics and test set metrics will be used to compare models.

4.4.3 Sliding Window Normalization

The timeseries are non-stationary, which is apparent from simple visual inspection (for more timeseries, see Section 7). Therefore, a variant of a sliding window moving average transformation [47] is proposed. The goal is to remove large trends by dividing the existing time series using simple moving averages. Since the data is highly intermittent, it can be tricky to scale; therefore, the moving average of the IPs has also be included. The normalization factor for scaling is defined as

$$\text{norm\_factor}_t = \frac{1}{2}\left(\sum_{i=t-12}^{t} \frac{y_i}{12} + \sum_{i=t}^{t+12} \frac{x_{t,i}}{12}\right), \tag{21}$$

where $x_{t,i}$ is the planned implementation at time t for future time i. The normalized values are then simply divided by the norm factor

$$y_t^{norm} = \frac{y_t}{\text{norm\_factor}_t}.$$

Notable is that this method does not remove variability from the model. The choice of scaling the data is to remove scale between timeseries which the models would otherwise have to learn. Moreover, this removes changes in trend in the timeseries. For instance, the segment shown in Figure 12 has a large peak during 2023 and otherwise relatively low volumes. By normalizing the data, the model can learn the changes in data and does not need to learn the changes in scale. This is analogous to differencing non-stationary datasets. The choice of normalizing instead of differencing is to be able to take into account IPs which are good indicators of changes in scale. Moreover, there is a risk that models will predict differences incorrectly when there is a large change in scale and therefore all predictions after that will be largely incorrect.

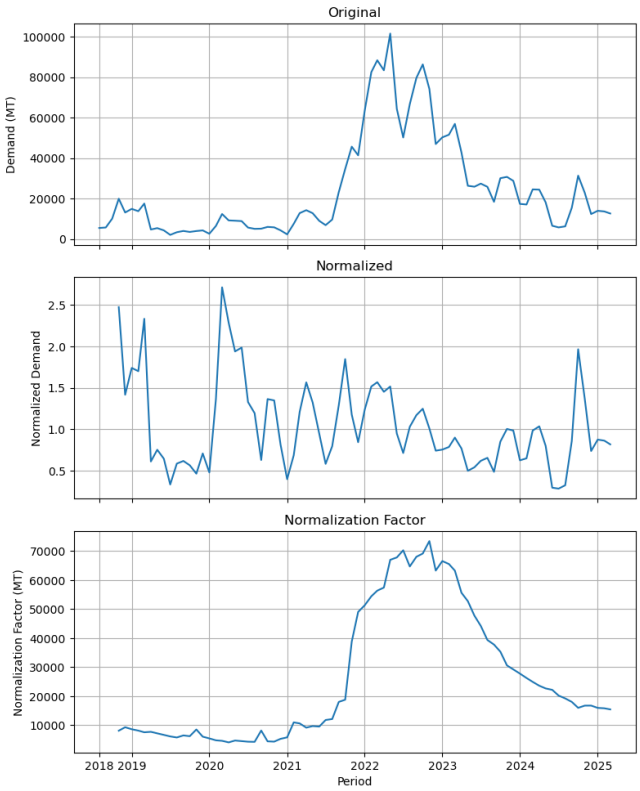


Figure 12: Example of a segment normalization transformation, removing the trend.

For some cases, the scaling is not possible at training; these timeseries will not be trained on. For inference, if normalization is not possible, meaning there are no historical values in the past 12 months and no plans, the predictions are set to zero.

4.4.4 Loss Function

The selection of an appropriate loss function is a critical step in model development, as it directly guides the optimization process during training [35]. Given the different characteristics of the raw and normalized data in this study, two distinct loss functions were chosen for the LightGBM models. For the raw model, the standard RMSE was used as a loss function, this closely resembles the evaluation criteria, optimizing for the mean. However, RMSE is sensitive to large errors, which can disproportionately influence model training when applied to data with a heavy-tailed distribution, as is the case here. To address this sensitivity to outliers, a different strategy was employed for the models trained on the normalized data. For these models, the Huber loss (see Equation (15)) function was selected. The normalization process brings all time series to a comparable scale, which makes it possible to use a single, meaningful δ value for the Huber loss across all segments. This offers a distinct advantage, as applying Huber loss to the raw, unscaled data would require a different delta for each time series.

4.5 Evaluation

For evaluation, we are concerned with the accuracy of the model as well as the bias. Bias is important since models can sometimes systematically under- or overestimate values, which is unwanted. A cumulative bias is proposed

$$\text{CBIASP} = \frac{\sum_{s \in S} |\text{BIAS}_s|}{\bar{y}},$$

where $\bar{y} = \frac{1}{|S|} \sum_{s \in S} \frac{1}{h} \sum_{t=T}^{T+h} y_{s,t}$ is the mean of the target over all segments and $|S|$ is the count of all segments.

However, accuracy is also an important metric; therefore, a score that combines accuracy and bias has been proposed as the main evaluation metric. For forecasting intermittent demand we are concerned with optimizing for the mean and not median, therefore a variant of RMSE is wanted. Since the dataset contains multiple time series a scaled or percentage-based error metric should be used for comparisons between time series. Moreover, the time series exhibit intermittent demand patterns with multiple zero values. This makes percentage-based evaluation metrics unstable due to division by zero. Therefore RMSSE (see Section 3.6.1) will be used.

The aforementioned metrics are for single time series in the dataset; however, these should be combined for an overall score. Similarly to Makridakis, Spiliotis, and Assimakopoulos [18], this thesis proposes using a weighted sum of the metrics. A weighting based on historical volume is proposed

$$w_s = \frac{\sum_{t=T-12}^{T} y_{s,t}}{12}, \tag{22}$$

where w_s is the rolling 12-month mean for the segment s. Yielding the total weighted accuracy of a prediction

$$\text{WRMSSE} = \sum_{s \in S} w_s RMSSE_s.$$

The inclusion of a weight metric not only gives importance to the current scale but can also be altered in the future. For this thesis, the total volume is optimized; however, the weight metric could in the future be changed to the dollar value of the commodity, since this is arguably more important than total volume. The dollar value of the commodities has not been taken into account as these data were not available for all commodities during the writing of this thesis. For hyperparameter optimization we want to tune the model based on one metric, for this we will define

$$\text{Score} = \text{WRMSSE} + \frac{1}{2}|\text{CBIASP}|, \tag{23}$$

weighting the bias metric by $1/2$ since ultimately the accuracy is more important, however the bias should still be taken into account. The absolute value of the bias is taken since values closer to zero are preferable.

Having more easily explainable metrics is also important for effective communication. Therefore MAE and relative MAE (rMAE) will be included for easy interpretation. Where

$$\text{rMAE} = \frac{\text{MAE}}{\bar{y}},$$

where $\bar{y}$ is the average target value for a specified period. Notable is that rMAE is not scaled per segment, but by the total demand during the evaluation period, this means that larger scaled timeseries will likely dominate the metric.

For probabilistic forecasts a MPL will be used defined in Equation (19), for a accuracy metric over all quantiles. We further define a weighted sum similar to [18], as

$$\text{WMPL} = \sum_{s \in S} w_s \text{MPL},$$

where w_s is the segment specific weight defined in 22.

Accuracy metrics give a quantitative sense of accuracy and bias. However, it is also important to do a visual evaluation of the models. This is not trivial, since the dataset contains approximately 800 timeseries. Therefore, segments will be sorted by accuracy and relative accuracy based on naive methods, after which the best and worst performing models will be analyzed. Much analysis could further be done, on total volume, volatility, characteristics of timeseries. For the worst and best performers, representative timeseries of different characteristics will be chosen, to get as close to a complete evaluation set as possible. Finally, as earlier stated, scaleups are a notable challenge, and difficult to predict. Therefore, analyzing datasets where the volume increases by many magnitudes in the test set will be visualized.

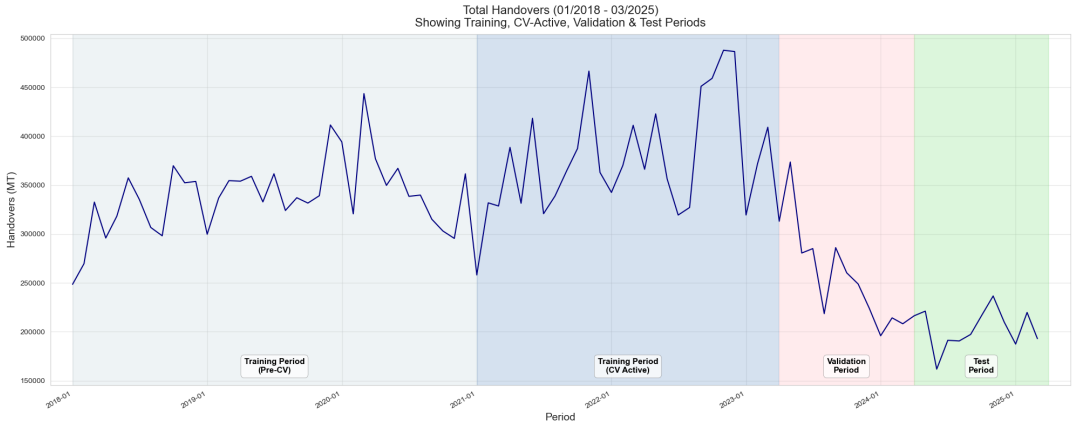


Figure 13: Date splits for train, validation and test.

This analysis includes a large amount of models, and using backtesting on all the models would be tedious and be practically too large of a scope for this thesis. Therefore, a choice has been made to do a initial evaluation on only the test set, from which the best naive, statistical and ML models will be included. This is also the case for the ML models, where in the initial round, hyperparameters will be tuned on

only the validation set (not using cross-validation). The sets have been visualized in Figure 13, where the initial train/validation/test split has been highlighted with different colors. The decision regarding a "two-staged" validation process and the "elimination" of certain models after the first stage is made for practical reasons. This approach aligns with the Design Science Research Methodology (DSRM) activities of Design & Development, Demonstration, and Evaluation, where models are constructed, tested, and assessed, followed by a cycle of adjustments. It is acknowledged that doing a initial test without cross-validation and eliminating some models which might perform poorly "by fluke" on the test set, and otherwise perform well. However, this is always the case for comparative analysis where all available methods cannot be tested, and therefore some methods have to be eliminated.

4.6 Robustness and Explainability

Per-design, backtesting gives a good overview of the stability of the model. Since we can analyze the models at different time points we see how this changes the model outputs. Moreover, since the dataset has a wide variety of different timeseries, a visual analysis of different predictions is a good way to get a sense of stability. For visual inspection, it is important to choose the timeseries in an unbiased way and choosing a diverse type of timeseries. Moreover, the inclusion of multiple metrics furthermore reinforces the stability of the results, since sometimes one metric does not always capture all wanted behavior. Hyperparameter optimization will show if there is a significant deviation in metrics given shifts in parameters, and how much the change in a parameter affects the metrics. SHAP scores is a good way to analyze the importance of features and the robustness of the models. This can be done on a global level by including plots where all SHAP scores are represented, a common way for this is to use beeswarm plots [38, ch. 18].

4.7 Probabilistic Forecasting

A key aspect of modeling demand under volatile conditions, such as humanitarian demand is the uncertainty associated with the forecasts. Therefore, using LightGBM for modeling uncertainty will be demonstrated, albeit not as rigorously as the point forecasts. LightGBM can be developed into a probabilistic model by predicting quantiles. Quantiles can be predicted by setting the loss function to a pinball loss (PL) defined in Equation (18). This can be done by defining one LightGBM instance per quantile, which makes the model computationally heavier. To keep the scope of this thesis, we will demonstrate a probabilistic model based on the LightGBM direct; however, this can be generalized for the recursive model as well. Lagged values of quantiles where also added as additional features, shown in Section 7.

A naive moving average model will be used for comparison where the quantiles will be derived from a Gaussian distribution based on historical standard deviation from the past 24 months. To make comparisons more meaningful, another method which uses LightGBM direct median forecast and normal distribution for quantiles will also be employed. To get a more meaningful sense of the uncertainty for each

forecast, the standard deviation of the residuals from the UIP model will be used. The choice of using UIP residuals is since these can easily be computed, another possible option could be to use the residuals from the LightGBM models from cross-validation. This would be a computationally heavier solution and have higher inaccuracies for the earlier cross-validation folds as the training set is smaller. For these reasons, the LightGBM residuals were not used. Finally, since the LightGBM direct upper quantiles can be somewhat unstable, these will be smoothed with a 7 month rolling average. All models will be compared based on WMPL and coverage ratios.

4.8 Software and Library Versions

The experiments were conducted primarily using Jupyter notebooks with Python. Git was used for version control, enabling the sharing of both the findings and the codebase with the HO. The models and analyses are implemented based on the following key software packages and their versions. The python packages used has been outlined in Table 3 and the hardware specifications in Table 4.

Table 3: Key software packages and their versions used in the experiments.

Package	**Version**
Python	3.12.2
Optuna	4.3.0
LightGBM	4.6.0
Statsmodels	0.14.4
NumPy	1.26.4
Pandas	2.2.3
Matplotlib	3.10.0
Pmdarima	2.0.4

Table 4: Hardware specifications of the system used for experiments

Component	**Specification**
Computer Model	MacBook Pro
Processor	Apple M4 Pro
Total Cores	*12 (8 performance and 4 efficiency)*
Memory (RAM)	24 GB
Operating System	macOS Sequoia 15.5

5 Results

This section introduces the results from the case study as described in Section 4. This section is split into two subsections: the point forecast and probabilistic forecasts. A large emphasis is placed on the point forecasts, while the probabilistic forecasts serve to demonstrate the model's generalizability for predicting uncertainty. Accordingly, methods such as backtesting and cross-validation are applied exclusively to the point forecasts.

5.1 Point Forecast

For the point forecasts, a two-staged evaluation process was implemented, where models showing promise in the initial round were chosen on a single test set. For the second round, chosen models were backtested and ML models tuned with cross-validation.

5.1.1 Initial Evaluation Runs

For the initial model runs, all models were fitted and evaluated on the test set. The ML models were tuned on the validation set with 50 trial runs, optimizing for the score defined in Equation (23). The final models where trained on the train and validation set. The eight hyperparameters were n_estimators, learning_rate, num_leaves, max_depth, min_data_in_leaf, feature_fraction, lambda_l1 and lambda_l2. The tunable ranges are specified in Section 7. The statistical and naive methods were fitted on the train and validation set and tested on the test set.

The results are presented in Table 5, note that for all metrics a lower number is preferred. The implementation plans (IPs) performed poorly, achieving a higher (worse) score than any other model except for Auto-ARIMA. This underperformance is partly due to the large positive bias, which also leads to higher WRMSSE and rMAE metrics. The unbiased implementation plan (UIP) models are the modified IP with the bias removed via a rolling average bias ratio. These performed significantly better, especially UIP12, outperforming all models except the LightGBM models. As expected, the UIP models also had low bias. Interestingly, MA3 had the least bias, even less than the UIP models; however, when looking at MAE and the score metrics, MA6 outperformed all other MA models. The exponential smoothing model (ES) did not perform well, although better than the IP. The LightGBM models performed well. Notable is the larger rMAE on the LightGBM raw models. The lowest score was achieved by the LightGBM Recursive Raw model. However, with a considerably higher rMAE, this points to the model working well on variable timeseries and having a small bias. The LightGBM direct norm had the best MAE, outperforming the next best model by over 10MT. The worst performing model was Auto-ARIMA, which was the only model that performed worse than the IPs.

Table 5: Key performance metrics of forecasting models on the test set.

Model	Score	WRMSSE	CBIASP	rMAE
IP	1.61	1.30	0.62	1.03
ES	1.32	1.15	0.34	0.83
Auto-ARIMA	2.29	1.84	0.90	1.44
UIP3	0.99	0.92	-0.14	0.99
UIP6	0.92	0.89	-0.06	0.92
UIP12	0.89	0.88	-0.02	0.89
MA3	0.97	0.97	**0.01**	0.73
MA6	0.95	0.90	0.10	0.67
MA12	1.14	1.01	0.26	0.80
LightGBM Direct Norm	0.93	0.79	-0.28	**0.60**
LightGBM Direct Raw	1.03	0.85	-0.35	0.72
LightGBM Recursive Norm	0.91	**0.77**	-0.29	0.62
LightGBM Recursive Raw	**0.83**	0.81	-0.04	0.75

Based on these findings, the LightGBM normalized models, UIP6. MA6, ES and IP, will be further compared via backtesting and tuned in cross-validation. These eliminate a large portion of the models while still maintaining a good mix of ML, statistical, naive and judgmental models for a fair comparison. A choice was made to not include the LightGBM recursive raw, despite its low score, since the rMAE was higher than MA and UIP. UIP12 and MA6 will be included in the analysis, since they had the best performances out of their respective versions. Auto-ARIMA will not be included due to its poor results.

5.1.2 Backtesting

For the backtesting runs, the LightGBM normalized models are tuned in an expanding window cross-validation fashion. The number of tuned parameters are also shrunk by keeping the hyperparameters that have a low importance and tuning the most important parameters. For both the direct and recursive LightGBM the learning_rate and n_estimators were tuned. This allowed for faster iterations which led to more folds. Taking an average over model predictions is a common ensembling approach that can increase accuracy [15], as was the case in the M5 competition [18]. Therefore, this study will test ensembles created by averaging the predictions from the LightGBM and UIP models. Specifically, variations between the LightGBM models and the UIP6 (hereafter referred to as UIP) will be evaluated.

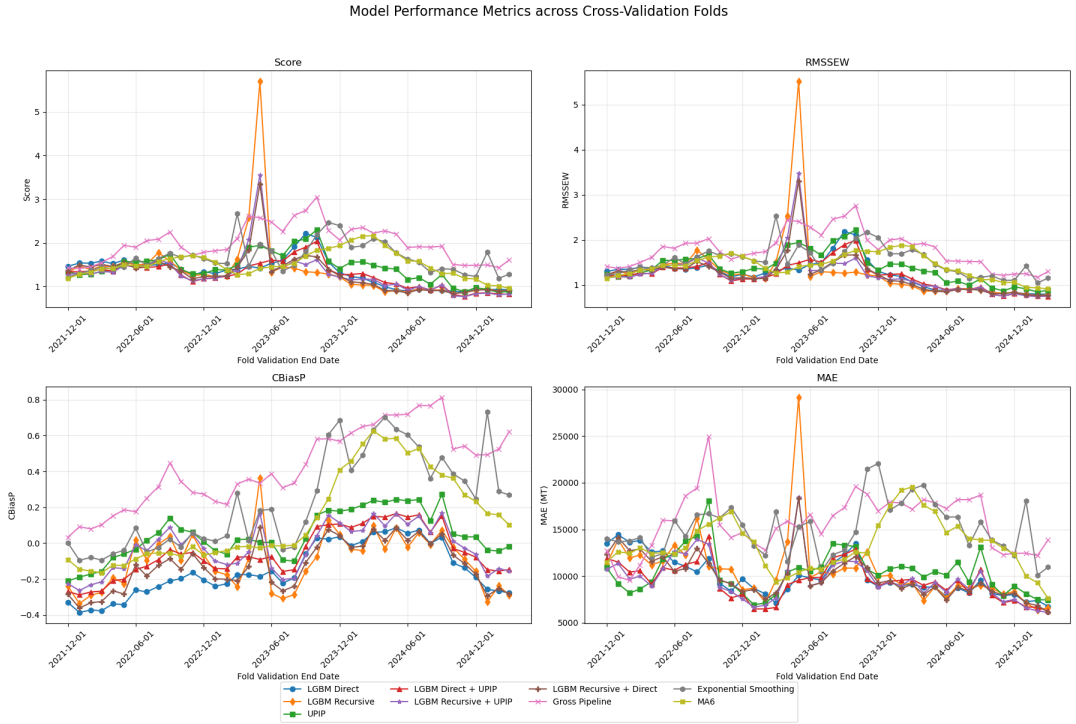


Figure 14: Backtesting Metrics for Each Fold.

Table 6: Average performance metrics in backtesting, metrics have been rounded to the two nearest decimals.

Model	Score	WRMSSE	rMAE	|CBIASP|	σ(Score)
IP	1.99	1.78	0.91	0.43	0.41
ES	1.66	1.53	0.79	0.26	0.35
MA6	1.52	1.42	0.72	0.20	**0.29**
UIP	1.43	1.40	0.70	**0.11**	0.34
LightGBM Dir.	1.33	1.24	**0.59**	0.18	0.33
LightGBM Rec.	1.37	1.30	0.62	0.15	0.77
LightGBM Dir. + UIP	**1.27**	**1.21**	0.61	0.13	0.30
LightGBM Rec. + UIP	1.29	1.23	0.62	0.11	0.46
LightGBM Rec. + Dir.	1.30	1.23	0.60	0.15	0.43

The backtesting results are visualized per fold in Figure 14, with aggregate metrics in Table 6. These findings are consistent with the initial test runs (Table 5), with the IP model performing worst on nearly all metrics except for the score's standard deviation, followed by the MA6 and UIP models. A key finding from the cross-validation was the instability of the recursive model. During fold 18, for instance, the model entered an unstable state with rapidly growing prediction errors (Figure 15). This behavior is quantified by its high standard deviation of the score, which was over twice that of the direct model. Despite this instability, the recursive model also showed unique

strengths. The recursive model was the only model that maintained a constant error during the downward trend in 2023, a period where all other models' errors increased.

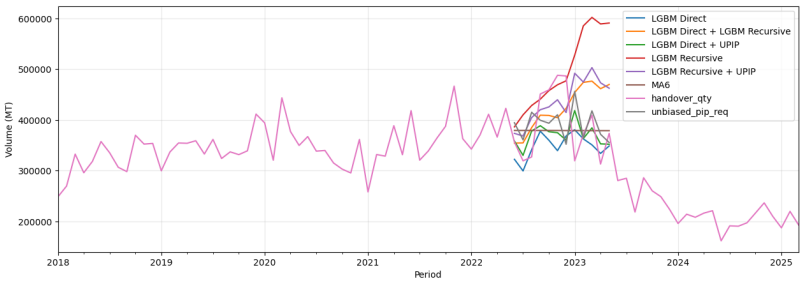


Figure 15: Aggregate Forecasts for cross-validation Fold 18.

The LightGBM direct and ensemble models performed well, where the combination of UIP and LightGBM direct had the best performance in score and WRMSSE. The direct model had a lower rMAE but higher bias, meaning the ensemble version (LightGBM direct+UIP) is correcting for this negative bias, slightly making the rMAE higher but lowering bias. From Figure 14 it is also clear that the high accuracies for MA6 is due to the relatively constant demand patterns during the test set, where MA6 is very inaccurate during the downwards trend in 2023. This is where the LightGBM model really outperforms UIP and MA6 considerably. During the test set, the differences are not as large. Another finding is that the LightGBM increase slightly as the training set increases; this is expected since most often a larger training set leads to better results.

Another important aspect of models is the stability. As stated earlier, the recursive LightGBM model has been deemed unstable due to erratic spikes in predictions. Visual inspection is a good way to identify large spikes in error. For a more quantitative approach, the standard deviation of the score has also been included. MA6 had the most stable score overall; the LightGBM direct + UIP came in at a close second, only having a standard deviation of 0.01 more than MA6.

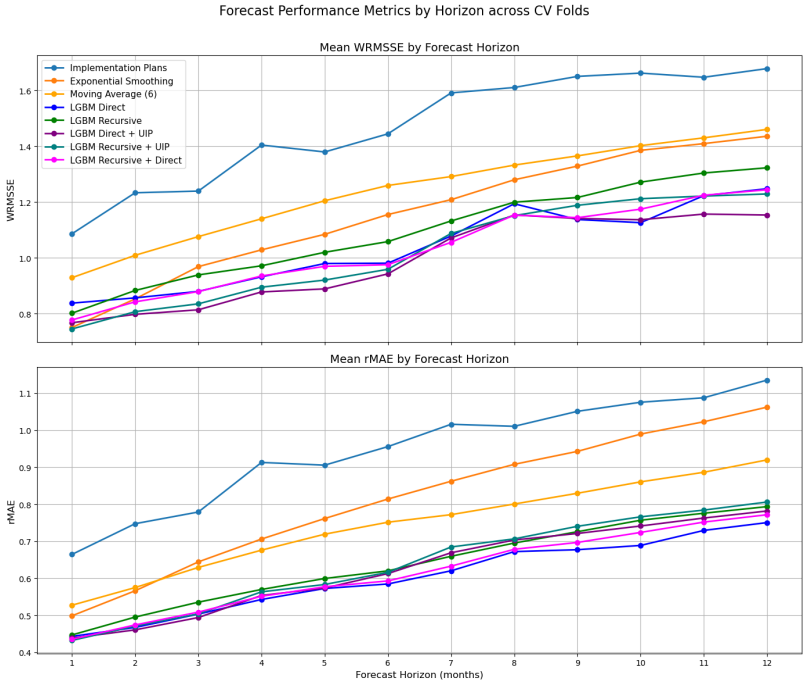


Figure 16: Average WRMSSE and rMAE per forecast horizon over backtesting folds.

The models were also evaluated over each forecast horizon within the backtesting folds. The rMAE and WRMSSE are presented for all 12 horizons in Figure 16. As anticipated, the error tends to increase with the forecast horizon. There seems to be somewhat of a jump from three to four months looking at the IP and consequently UIP. All models are quite consistent in their order, where the LightGBM direct has the lowest rMAE over almost all horizons; however, for WRMSSE it is not as clear, and LightGBM direct seems to grow somewhat for horizons 11 and 12.

The final aggregate predictions are visualized in Figure 17. Here, the negative bias of the LightGBM models is evident; this is most likely due to the negative trend that preceded the period from which the models learned. Figure 17 also illustrates that relying on visual validation of aggregate predictions can be misleading, as both LightGBM direct and recursive models have more accurate predictions yet appear less accurate in the plot. Therefore, it is more useful to look at individual predictions.

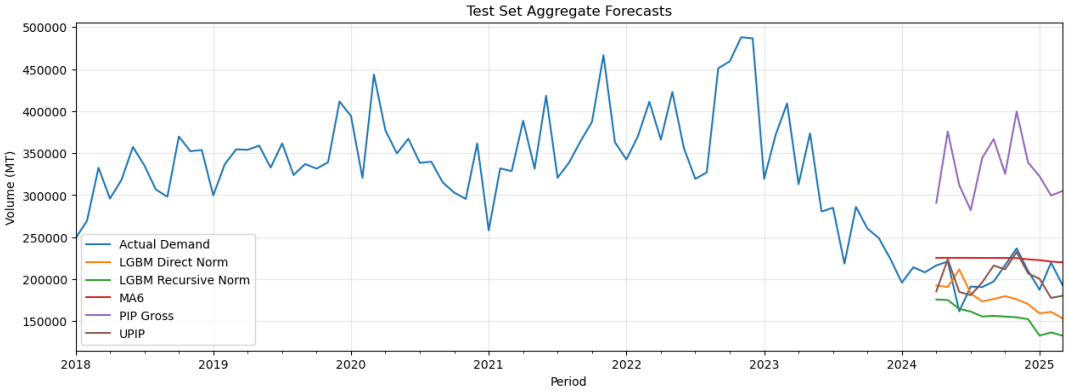


Figure 17: Aggregate Demand Predictions.

Figure 18 illustrates the better performing predictions of the LightGBM models. The top row displays the overall accuracy, while the bottom row presents accuracy relative to MA6. Conversely, Figure 19 showcases the least accurate predictions, again with the first row showing total accuracy and the second comparing it to UIP. Overall, the LightGBM models produce smooth predictions with mostly negative trends. This counteracts the sporadic nature of IPs. This is to the detriment and benefit of the models, for instance (ISO21, CM42, AC12) in 19 the models incorrectly predict a large dip in demand, while in (ISO76, CM32, AC17) in 18 it does the same correctly. The downwards predictions are most likely due to the historical negative trends, and therefore it might be a chance that the models sometimes predict these negative trends correctly in the test set.

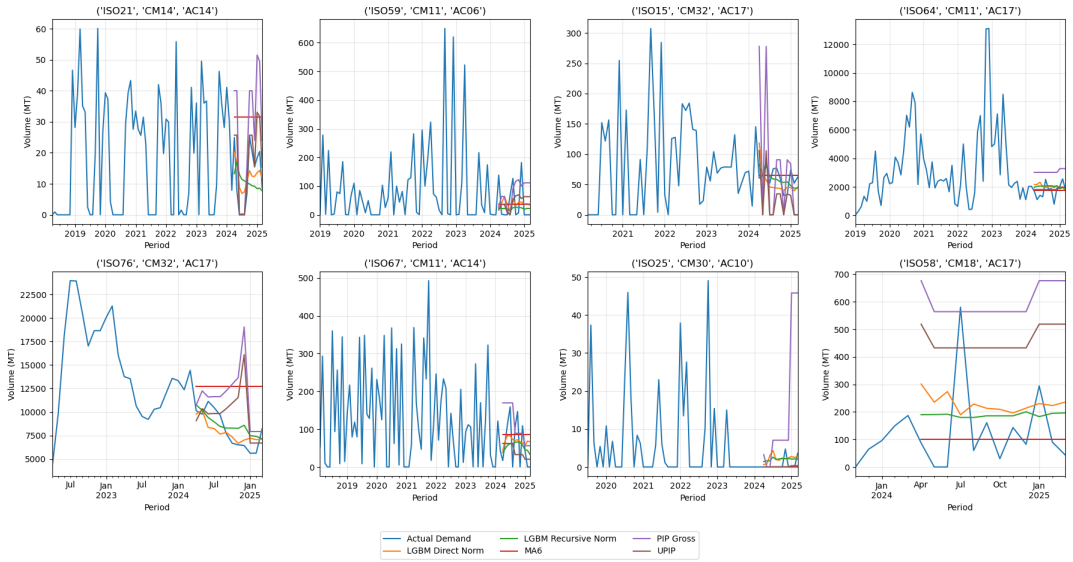


Figure 18: Examples of accurate LightGBM predictions, first row has high accuracies and second row has high accuracy compared to UIP.

When analyzing the relative low accuracy, IPs have mostly predicted large spikes in demand The LightGBM models are conservative, meaning they do not predict large

spikes even if there is a spike in IPs. This can also increase accuracy, when analyzing the best predictions, for instance in ("ISO25", "CM30", "AC10"). In this sense the LightGBM models more closely resemble moving average models or statistical models. There are also examples of large increases in demand, notable in Figure 19 where large spikes in demand occur in the test set. These large increases in demand are visible in the IPs which is why the LightGBM models can pick up on the large increases in demand. Here it is also apparent where the moving average and other statistical models do not perform well. Overall analyzing metrics and predictions show that the LightGBM models increase the accuracy of predictions, however for some timeseries they perform considerably worse than the naive models. This might be due to the earlier negative trend; with more training data the models could perform better.

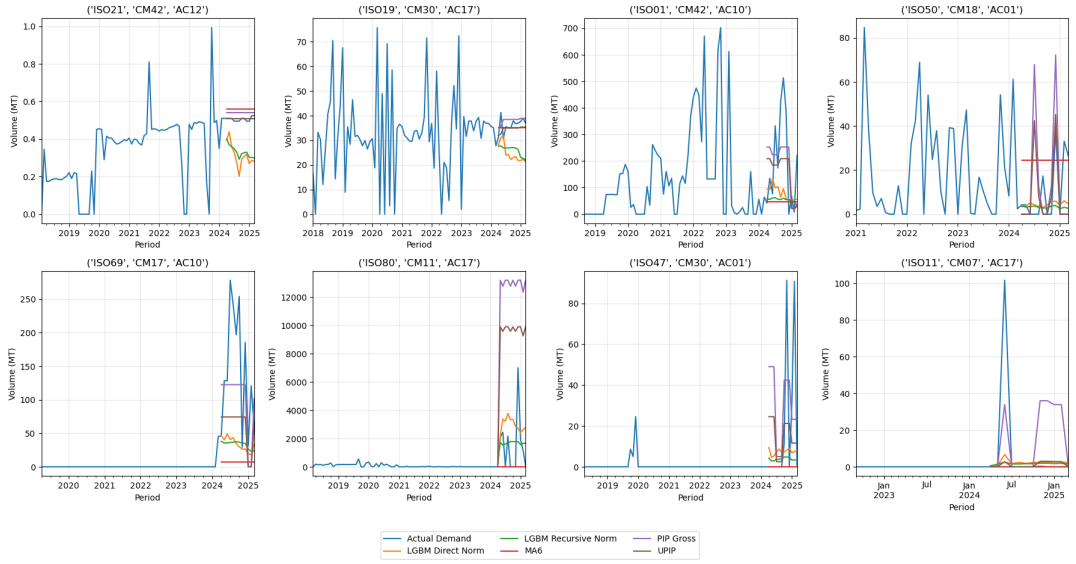


Figure 19: Examples from inaccurate LightGBM direct predictions, first row has low accuracies and second row has low accuracy compared to UIP.

The largest errors were in timeseries with historically very low demand with sudden increases in demand (see second row of Figure 19). It is not expected for models to have good results for these kinds of events since they are inherently very difficult to predict. One takeaway is that the LightGBM models use the IP data, however the models are very conservative and predict under the IP or UIP. This leads to better predictions for (ISO80, CM11, AC17) but worse in (ISO69, CM17, AC10), as compared to UIP and IP. As expected, the MA6 performs very poorly on these. (ISO11, CM07, AC17) is also interesting in the sense that the models being conservative leads to a large under-prediction for the actualized demand spike, however a much lower error for the period where the IP is large in the end of 2024.

Finally, a two-way permutation test is performed between absolute errors and WRMSSE to see if there is a significant difference between MAE and WRMSSE. The choice has been made to do these separately instead of doing one permutation test on the scores, since this gives a more comprehensive overview of the differences

between individual metrics. The two-way permutation tests have been visualized in Figure 20. All the models have significant deviations from the IP, meaning that all the models outperformed the IP with statistical significance over 40 cross-validation folds. For MA6 it is a bit more nuanced, where only the LightGBM model has statistically significant deviations from MA6. Notably, a statistically significant difference between the recursive model was only seen in the recursive direct ensemble. This means that it is difficult to draw statistically significant conclusions between this model and others. This is most likely due to the instability of the model. However based on this, it is clear that the LightGBM direct model outperformed MA6, IP and UIP models with statistically significant differences in WRMSSE and MAE. The normalized LightGBM direct model seems to best predict food demand given the evaluation metrics and feature set, and accuracy can be further improved by using ensembles, however this change is not statistically significant.

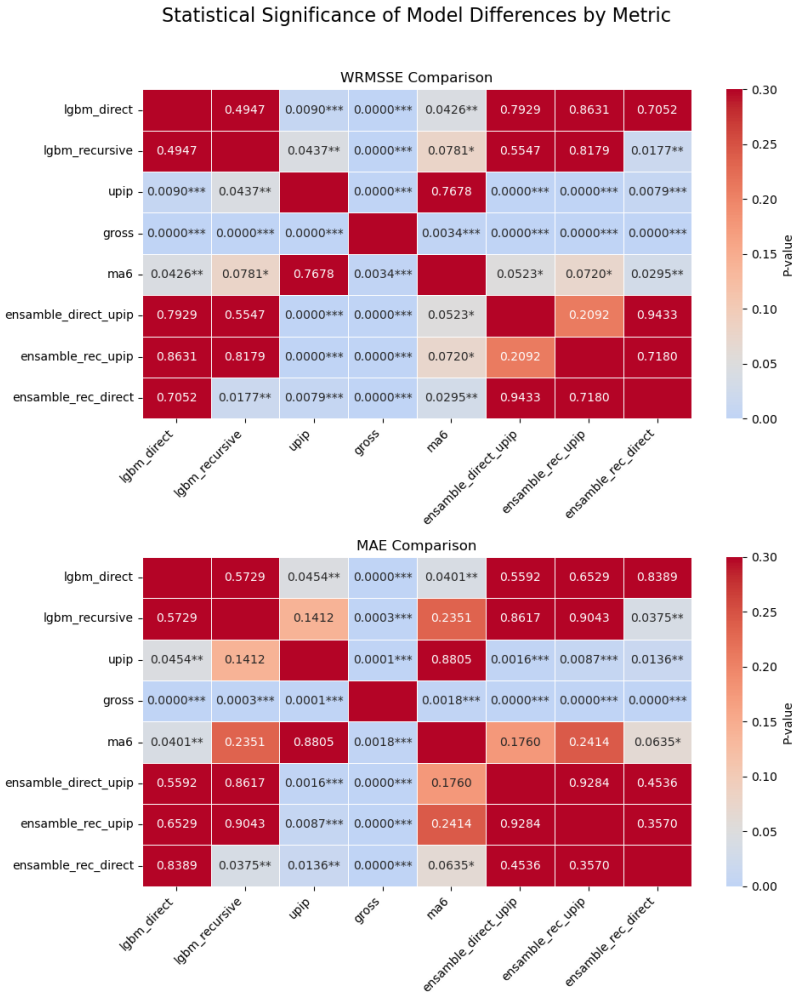


Figure 20: Two-way permutation on the differences between metrics for all segments and all cross-validation folds. Significance levels: p<0.1 -> *, p<0.01 -> **, p<0.1 -> *** (1.0×10^6 permutations).

5.1.3 Explainability

For the direct and recursive LightGBM models, the SHAP values were analyzed with beeswarm plots (Figures 21 and 22). The top features of the models are similar, where lagged features, as well as IP and inventory data were dominant. The lagged features are expected, especially lag 1 for the recursive model. Interestingly, resource_ratio was the most important feature for the direct model, while the fifth most important for the recursive one. The direct model has less emphasis on norm_IP_req than the recursive model. Interestingly, activities were more important than the country feature; this points to the importance of including activities in the model. The direct model used moving average values more while the recursive model relied on corporate level data such as expenditure data. This coincides with the recursive model performing better at picking up macro trends than the direct model. This is also visible in Figure 14 where the recursive model had good accuracy during the trend changes in 2023. A notable finding is that the recursive model uses the approximate expenditure on food feature. This could explain why it was able to predict the downwards trend for 2023. This could be due to a lower expenditure forecast for this year, which led to lower demand and was not picked up by the IPs.

The direct model picked up GDP growth where smaller GDP growth coincided with less demand. This is logical since GDP growth generally corresponds with the economic prosperity or stability of a country, which in turn leads to less need for aid. The recursive model used the inflation index more, where a larger inflation index corresponds with more demand. Since cost of living crises are one of the main drivers of food insecurity [5], this is expected. The IPC data [6] was wasn't highly important, most likely due to the small data set used, where a large portions of rows did not have IPC values. The resourced_ratio was the most important feature for the direct model and a very important feature for the recursive model and indicates that larger resource_ratios correspond to a larger demand. This makes sense, since if there is large stock then this will most likely be distributed in the future.

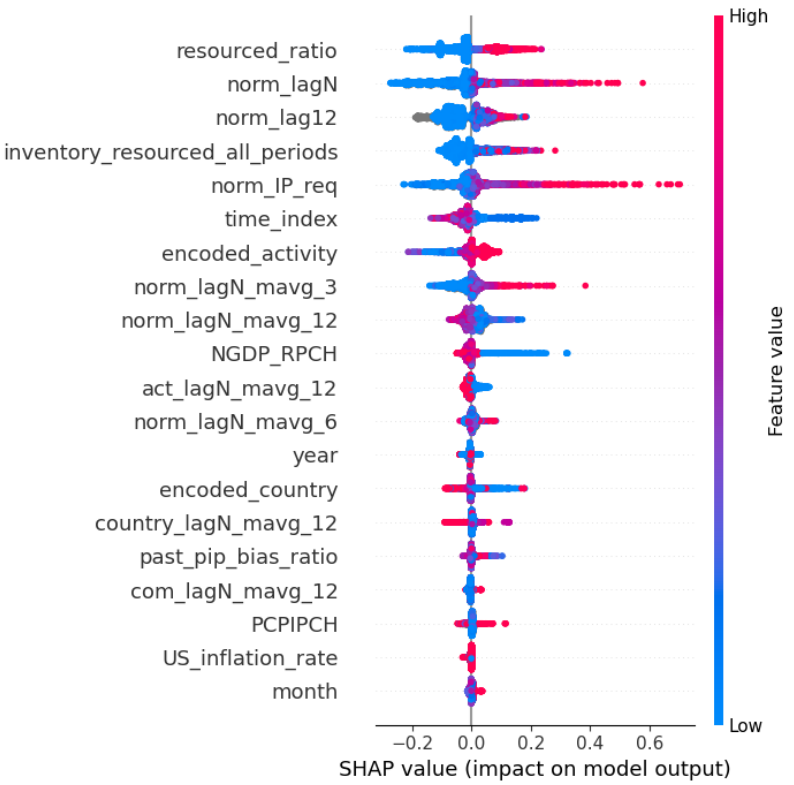


Figure 21: SHAP Values for LightGBM Direct.

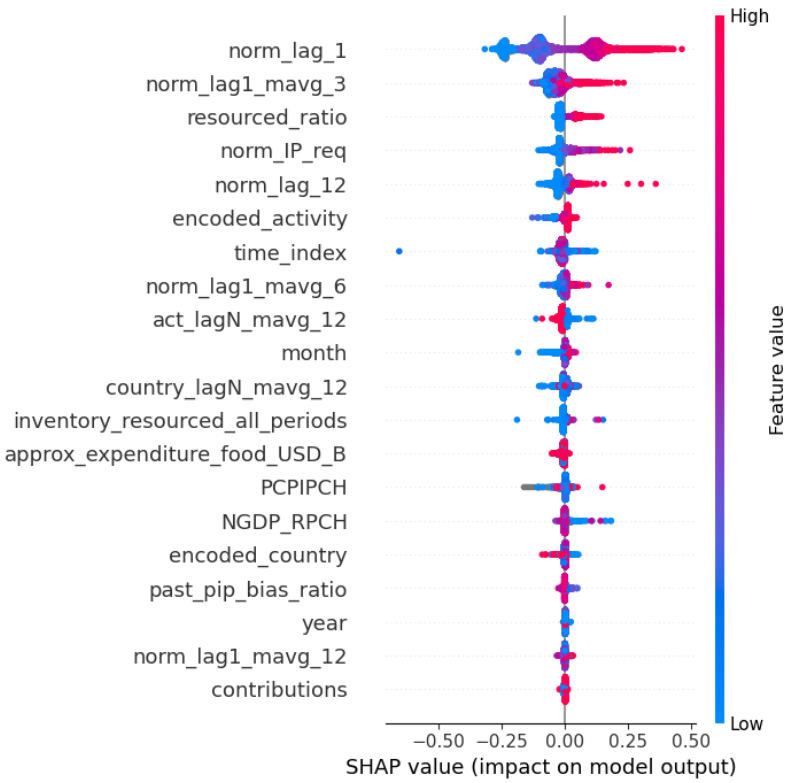


Figure 22: SHAP Values for LightGBM Recursive.

5.2 Probabilistic Forecasts

For the probabilistic forecast, 5 quantiles are predicted: the 10%, 25%, 50%, 75%, and 90%. A separate LightGBM Direct Norm model was fitted on each of these quantiles, and the accuracy was measured by taking the average pinball loss. The coverage of the quantiles was also reported. A 6 month moving average model (MA6) is also shown for comparison. The MA6 model estimates the mean which is the 50% quantile, the rest of the quantiles are estimated based on a Gaussian distribution from the historical residuals.

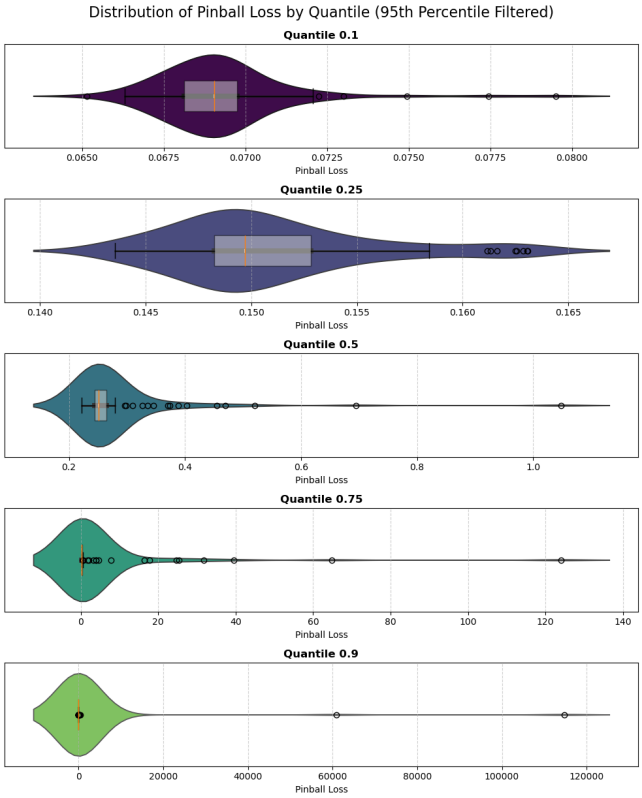


Figure 23: Distribution of WMPL over Optuna trial runs, 95% percentile removed for readability.

Initially, a model was tried where the hyperparameters for each quantile model were the same. This proved to fail, since the larger quantiles are considerably more unstable. This is expected due to the large right skew of the data, with a sparse and long right tail. Therefore, the hyperparameters were tuned individually for each quantile. This was done with 30 trials per quantile, and only on the validation set. The uncertainty levels are still volatile; therefore, a 7 month rolling average was done in postprocessing, which yielded better results. The Optuna WMPL scores over all trial runs are shown in Figure 23. It is evident that certain hyperparameter combinations result in errors that are orders of magnitude larger than the average error.

The results are shown in Table 7, where the LightGBM model clearly outperforms the MA6 naive model. The smoothed model has the strongest performance, especially in the higher quantiles where the coverage ratio was almost exactly equal to the quantiles for the 75% and 90% percentile. This is also reflected in the WMPL. The smoothed model had across the board stronger results than the standard direct LightGBM model. However, both of these had poor performance on the lower quantiles where the MA outperformed all models on the 25% quantile and the LightGBM naive model on the 10% quantile. This points to the lower quantiles following more of a normal distribution while the upper quantiles are different. This is expected since the data is heavily skewed to the right.

	MA6 Gaussian	Gaussian	Direct	Direct Smoothed
WMPL	0.1912	0.1719	0.1620	**0.1560**
Coverage Ratios				
$\tau = 0.1$	0.132	**0.096**	0.082	0.084
$\tau = 0.25$	**0.255**	0.137	0.195	0.192
$\tau = 0.5$	0.697	0.403	0.403	**0.409**
$\tau = 0.75$	0.863	0.773	0.740	**0.749**
$\tau = 0.9$	0.936	0.912	0.894	**0.901**

Table 7: Comparison of Weighted Mean Pinball Loss (WMPL) and Coverage Ratios for different probabilistic models. The target quantile is denoted by τ. (All models are normalized LightGBM models expect MA6.)

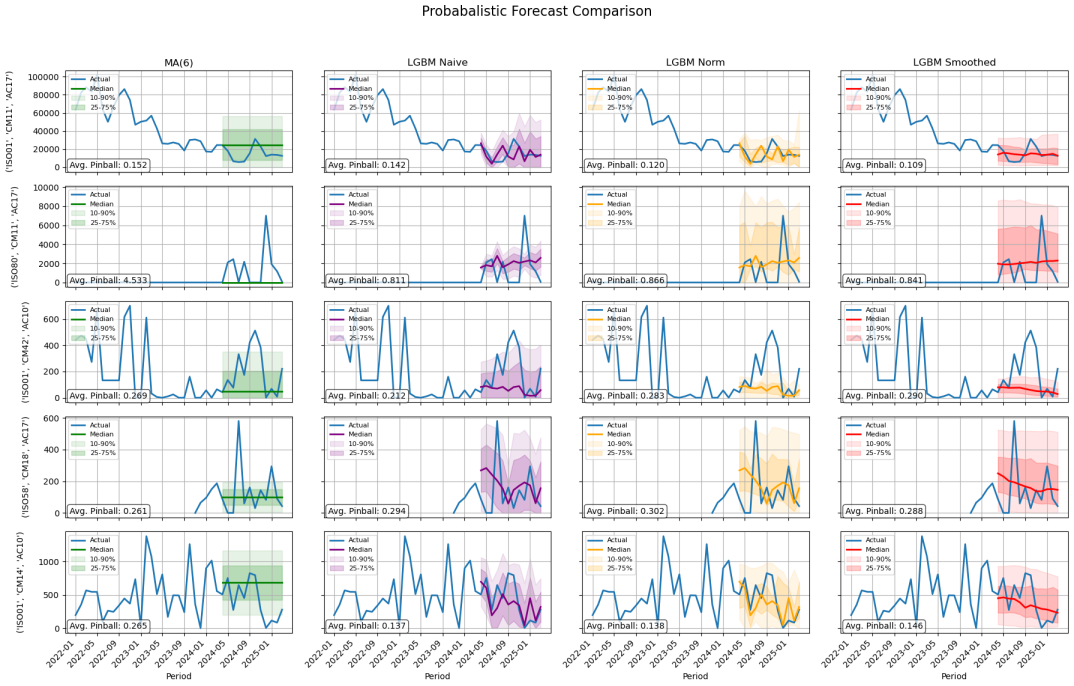


Figure 24: Probabilistic forecast examples, and different variants of Normalized LightGBM Direct.

Some results have been visualized in Figure 24 where one can see the strength of different versions of the model. Most notable are the sudden spikes in demand, where LightGBM models considerably outperform MA, and would most likely outperform all models which do not include exogenous data points. A permutation test was performed on the models seen in Figure 25. It shows that most WMPL findings were significant, except between the LightGBM naive and MA6. This might be due to the similar uncertainty intervals of the models, both assuming a Gaussian distribution. Moreover, since the data is quite constant for the test set, the LightGBM median forecasts could be quite close to MA6 predictions.

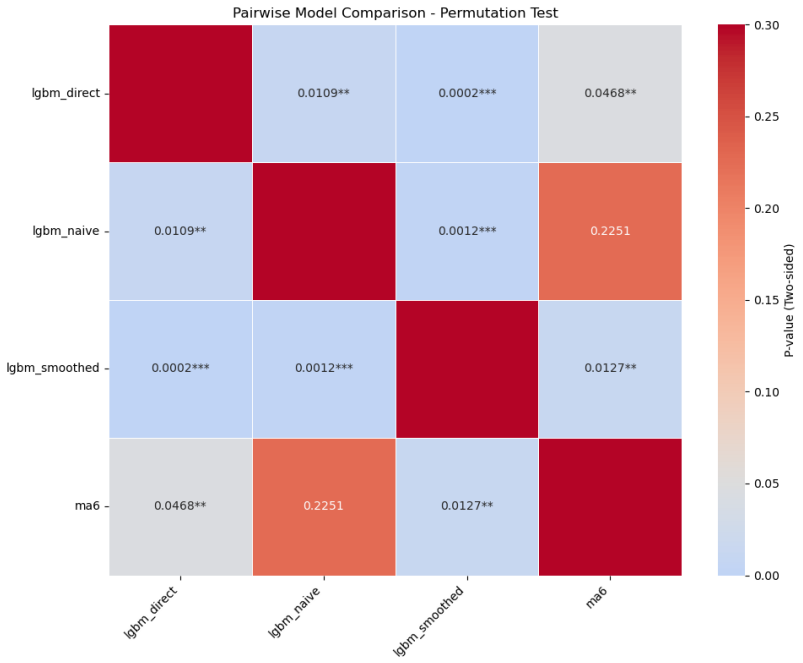


Figure 25: Two-way permutation between segment level WMPL. Significance levels: p<0.1 -> *, p<0.01 -> **, p<0.1 -> *** (1.0×10^6 permutations).

6 Discussion

This thesis expands upon the scarce literature on demand forecasting within the humanitarian sector [3]. As stated by Babai, Boylan, and Rostami-Tabar [3], there exists no current literature within humanitarian demand forecasting, and based on the literature review we found this to be true. We see this thesis as a contribution to this area of research, which is crucial for more efficient supply chain management [3]. During the writing of this thesis, Ouchtar, t'Serstevens, and Rahman [27] published a report analyzing statistical demand forecasting methods for a smaller NGO. We see this thesis as a complementary study compared to that one, by applying similar methods to a larger HO, and further analyzing the impact of using ML models.

This section discusses the findings presented in Section 5, interpreting their significance in relation to the research questions defined in Section 1. It will first assess the performance of statistical, naive and judgmental methods (RQ1), then evaluate the accuracy improvements and model variations of LightGBM (RQ2), analyze the reliability and interpretability of the proposed models (RQ3), and finally, reflect on the effectiveness of using LightGBM for characterizing demand uncertainty (RQ4).

6.1 Accuracy of Statistical Methods

Part of this thesis was to study the accuracy of naive, statistical and judgmental models in predicting demand. By first testing exponential smoothing, Auto-ARIMA and three different moving averages, it was found that Auto-ARIMA performed very badly compared to all other models. This is in line with the findings by Ouchtar, t'Serstevens, and Rahman [27] where Auto-ARIMA was the worst performing model. The exponential smoothing did not perform well either, but had better predictions than the country office implementation plans (IPs). All moving averages outperformed the IPs and statistical models, with the six month moving average having the best performance. Moreover, the IPs had considerable positive bias, therefore a simple unbiased version was tested where the bias was calculated in a rolling fashion and removed from the IPs. This greatly increased the accuracy, and it was found that a 12 month rolling bias ratio had the best performance. Based on this, the unbiased implementation plan (UIP), six-month moving average (MA6), IPs and exponential smoothing were backtested.

In backtesting, the UIP, MA, ES and IP models performed worse on average compared with the test set. This is most likely due to changes in trends which are very hard to predict by just fitting on historical values. The UIP slightly outperformed all models, with the MA scoring higher across the board than ES and IP. The IP was the worst performer, exhibiting a large positive bias, high standard deviation, and higher rMAE and WRMSSE. The accuracy also decreased with a higher forecast horizon, which is expected.

The poor performance of these statistical models is likely a direct result of the data's high volatility. Models such as ARIMA and ES are best suited for time series with more regular and predictable patterns. Given that only about 4% of the demand data in this study is classified as 'smooth' while the majority is 'lumpy', the weak

performance of these traditional models is expected. This finding underscores the need to apply more advanced methods, like the gradient boosting decision trees evaluated in this thesis, which are better equipped to handle such complex and erratic data [18]. Moreover, this underscores the power of simple methods, such as bias removal of IPs or simply using moving average models as opposed to statistical methods. However, these simpler methods have their own critical limitation: being purely fit on historical data, they are not suited for all situations, particularly during significant trend changes. This suggests that if an organization were to rely solely on naive or simple statistical models, these would still need to be paired with expert judgmental forecasting to anticipate future shifts not present in the historical data.

6.2 LightGBM Models to Improve Accuracy

Due to the difficult nature of the timeseries, LightGBM models were implemented. These were compared to the statistical and naive methods for benchmarking. Initially four different versions were tested, a recursive and direct version on normalized and raw data. Based on initial runs on the test set, the normalized methods outperformed the raw ones, therefore these methods were further tuned with cross-validation and tested with backtesting. This was done to get a good sense of how the models performed over multiple different time periods as well as robust tuning on multiple different evaluation windows.

Overall, the LightGBM models performed well, notable was the normalized variant of the LightGBM Direct version, which had the lowest MAE and WRMSSE. This was despite critical data features such as funding data which could probably improve the accuracy even further. In backtesting the LightGBM direct outperformed the naive, statistical and UIP models with a 7% decrease in score and a 16% decrease in MAE, compared to the best baseline models. Moreover, these differences where statistically significant. Instability was found in the recursive LightGBM models, both in cross-validation and in hyperparameter tuning (see Section 7), however the models where able to accurately predict changes in trends when all other models had higher errors. This is in line with Makridakis, Spiliotis, and Assimakopoulos [18] where it was found that recursive LightGBM versions had high accuracies but were unstable. The direct model exhibited good stability with a low standard deviation of the score, this stability was further increased in the ensemble versions.

It has been shown that taking the average predictions of different models can yield gains in accuracy [15], [18]. Therefore, simple averages of different model combinations were tested; this led to positive results, albeit not statistically significant differences between standard LightGBM models and ensemble versions. The highest scores were found by combining the LightGBM normalized direct model with the unbiased 12-month IPs. From these findings, there are statistically significant differences in multiple accuracy metrics as opposed to the IPs. Moreover, LightGBM models outperform unbiased versions of the IPs, moving averages, and statistical models such as ARIMA and ES, in all metrics except for bias.

The trend removal, i.e. rolling normalization of the data, was found to work well. This removed large changes in trends, allowing models to predict relative changes in

the data. This led to considerable accuracy gains in LightGBM models; however, there was a higher negative bias. The negative bias is a noteworthy finding, as systematic underprediction is an unwanted feature of the models. Typically, within humanitarian demand, a slight overstock is generally preferred, especially for non-perishable items that do not expire as quickly. Consequently, this should be accounted for in post-processing and demand planning if these models were to be taken into production. This negative bias is likely due to the negative trend in the test sets, suggesting the model optimizes for this. The negative trend arises from smaller amounts of funding; hence, if funding data were to be considered, it could offset some of this "systematic" negative bias.

The models were trained on a broad feature set. The most influential features were inventory data and IPs, as well as lagged demand and lagged moving averages. Most of the features had logical effect on the models; for instance, a higher resourced_ratio led to more demand, i.e. the more commodities that are stored, the more aid is distributed. Additionally, IPs were in line with demand, where higher plans did indeed lead to higher predicted demand in the models. Other notable features were that GDP growth had a negative effect on demand, indicating that more economic growth corresponds to less need for food aid. The IPC data did not have a notable effect on food demand; this is most likely due to the data populating a bit less than half of the rows. Moreover, since the IPC data consisted of five different features, their combined importance might be considerable even if the individual features contributed small amounts.

6.3 LightGBM for Modeling Uncertainty

To demonstrate the use of LightGBM in probabilistic forecasting, benchmarking was conducted on the test set. This was done by fitting a separate LightGBM instance per each quantile by fitting to the corresponding pinball loss function. Four different models were employed for benchmarking, with a six-month moving average (MA6) model utilizing a Gaussian distribution for quantiles as the naive benchmark model. To ensure a more fair comparison, a LightGBM model predicting the median and using a Gaussian distribution for modeling quantiles was also utilized. Finally, a LightGBM model fitted on each quantile was tested; to ensure stability, a smoothed version of this model was also implemented.

For the LightGBM direct model, the weighted mean pinball loss (WMPL) was over 15% lower compared to the MA6, with this reduction being statistically significant. Moreover, smoothing out the LightGBM predictions further decreased the WMPL to 0.156 compared to the MA6 0.191. It also had more accurate coverage ratios for all quantiles expect the 25% where there was a noticeably larger deviation in the LightGBM model. A drawback for the LightGBM model was its stability, especially when modeling the higher quantiles. These models were highly unstable, which was visible in large errors when changing parameters in hyperparameter optimization. Ultimately, while LightGBM models effectively capture demand uncertainty better than naive models, they exhibit instability. As a result, tuning or further development in ensuring robustness are essential for practical application. Moreover, the lower quantiles were quite inaccurate and a simple moving average with

a Gaussian distribution modeled these better on average when looking at coverage ratios. This paired with the instability in the models points towards more development being needed for taking these models into practice; however, the early developments are promising. For now, simply modeling the median and inferring the quantiles from a Gaussian distribution seems to deliver more robust results; though more inaccurate results. Therefore, a more sophisticated distributional assumption could deliver more robust and accurate results. Since the data is for the most part lumpy, intermittent, and erratic, distributional assumptions such as negative binomial could provide better results. Moreover, one could use parametric regression models such as LightGBMLSS [46] to directly fit to these distributions; this would generate more robust models.

6.4 Limitations

This thesis seeks to develop forecasting models for a complex problem; therefore, there are simplifying assumptions and limitations in the study. One major limitation is that adding more data should be done in the future to increase the model accuracy further, for instance funding forecasts. This will lead to the models behaving differently, which might require retuning and new benchmarking to be done for optimal models. Another limitation is that while we can confidently rule that LightGBM models perform better than statistical methods such as ES and Auto-ARIMA, no tests were done on other ML models. For instance Herteux, Raeth, Martini, *et al.* [16] found evidence that Reserve Computing can outperform GBDT models in predicting short-term food insecurity.

While statistical significance is tested, it is not the standard way of doing timeseries forecast comparisons. This was done for practical reasons due to the small sample size and large number of groupings. However, in longer horizons, a more meaningful test would be, for instance, the Diebold-Mariano test [15]. This would give a better analysis when measuring differences between forecasts; however, it is not trivial how this should be applied to grouped timeseries datasets as in this thesis.

A key assumption which was made is that the implementation plans (IPs) and inventory levels are independent of these forecasts. This assumption can be made as the primary use-case for this model is for global supply chain planning and inventory optimization. An obvious limitation is the meaningfulness and robustness of the LightGBM probabilistic model. This was as a proof of concept demonstrating that LightGBM models can improve probabilistic forecasts if measured against naive models. However, stability could further be improved with post-processing or trying other methods, such as parametric distributional models [46].

6.5 Avenues for Future Research

Building directly on the limitations identified in the previous section, several avenues for future research emerge. It is probable that gains in accuracy can be achieved by expanding the model feature set. For instance, more meaningful data on funding, climate, and food insecurity could further enhance accuracy. This could also lead to changes in the models and therefore parameter tuning and testing should be performed again with new feature sets.

The models developed in this thesis are optimized using volume-centric error metrics. A key avenue for future research is to create forecasts that are more directly aligned with financial and operational objectives by incorporating commodity price and criticality. One approach is to shift the optimization from physical volume to monetary value. This could be implemented by modifying the evaluation weighting scheme (see Equation (22)) to include commodity spot prices, or future prices if available. This would prioritize forecast accuracy for high-value items, supporting more cost-efficient procurement and inventory management. Alternatively, a humanitarian-impact approach could be adopted by weighting commodities based on their criticality. For example, specialized nutritional products might be prioritized over bulk grains, even if their volume or cost is lower. This would ensure that forecasts for life-saving items are the most accurate.

Some features are not available for segment level, for instance the macroeconomic indicators used in this thesis which are only available annually and for countries. These features could be better utilized by forecasting on higher aggregate levels such as the total demand per country instead of only segment level. Therefore, it could be beneficial to forecast on different hierarchy levels, combining these forecasts in post processing. Utilizing methods such as minimum trace reconciliation, one could combine these forecasts [31, ch. 11]. These types of methods have shown promise in complex hierarchical forecasting tasks and could be applied in this case as well [18].

Future work could also further study probabilistic forecasting methods. Since the current quantile forecasts are quite unstable, other methods such as residual based quantile estimates might yield more reliable predictions. Other models such as parametric models which assume some distribution and fit data to the parameters could be utilized. However, it is difficult to assume a certain distribution for one segment since these can differ, therefore one might need to first classify segments based on distributions after which the parameters of the distribution could be inferred.

Possible improvements in accuracy could further be gained by smoothing the data in preprocessing. For instance, a simple moving average could improve accuracy by reducing volatility in the training data. This would still serve its purpose for longer-term supply planning, since individual month predictions are not as important as the average demand over the neighboring months.

7 Conclusion

This thesis aimed to contribute to the limited literature on demand forecasting in the humanitarian sector. Forecasting demand enables improved supply chain management and, consequently, more efficient aid delivery. In this thesis, we applied naive, statistical, and machine learning (ML) demand forecasting models to predict the demand for WFP food commodities. Multiple different exogenous data points were included and different feature engineering methods to improve accuracy.

Four LightGBM variants were tested, direct and recursive with normalized and raw data. It was found that the normalized versions outperformed the raw ones. The normalized LightGBM recursive model was deemed unstable but had accurate predictions most of the time as compared to the benchmark methods. However, the LightGBM direct normalized model outperformed all other models metrics with a statistically significant degree. It was moreover found that combining judgmental and LightGBM models proved an effective way to remove some bias and increase the accuracy metrics of the LightGBM direct model.

For useful demand forecasts for supply planning, the uncertainty of the models should also be taken into account. Based on the point-forecast findings, the LightGBM model was expanded to predict quantiles, giving a sense of the uncertainty of the predictions. It was found that the LightGBM direct normalized models outperformed a naive normal MA6 model where the uncertainty was based on the standard deviation of historical residuals and assumed a Gaussian distribution. The upper quantiles were unstable and therefore a smoothing post-processing step was added which increased the accuracy further. Ultimately, the LightGBM normalized model performed better than the MA6 naive model based on WMPL measures.

The models implemented in this thesis are generalizable and allow for multiple new features and different parameters to be tested. Based on earlier literature and the lack of some key features, the models could most likely still be improved and developed on. Therefore, we see this as a starting point for more advanced ML based demand forecasting within the humanitarian sector.

References

[1] N. Altay and A. Narayanan, "Forecasting in humanitarian operations: Literature review and research needs", *International Journal of Forecasting*, vol. 38, no. 3, pp. 1234–1244, 2022.

[2] M. Moshtari, N. Altay, J. Heikkilä, and P. Gonçalves, "Procurement in humanitarian organizations: Body of knowledge and practitioner's challenges", *International Journal of Production Economics*, vol. 233, p. 108 017, 2021.

[3] M. Z. Babai, J. E. Boylan, and B. Rostami-Tabar, "Demand forecasting in supply chains: A review of aggregation and hierarchical approaches", *International Journal of Production Research*, vol. 60, no. 1, pp. 324–348, 2022.

[4] United Nations, *UN general assembly, transforming our world : The 2030 agenda for sustainable development*, 2015.

[5] WHO, FAO, IFAD, UNICEF, and WFP, "The state of food security and nutrition in the world 2024", FAO; IFAD; UNICEF; WFP; WHO; Rome, Italy, 2024.

[6] F. S. I. Network, "FSIN and global network against food crises 2024", Rome, Italy, 2024.

[7] R. M. Tomasini, *Humanitarian Logistics*, 1st ed. 2009. London: Palgrave Macmillan UK, 2009.

[8] United Nations General Assembly, *Strengthening of the coordination of emergency humanitarian assistance of the united nations*, 2003.

[9] E. V. D. Laan, M. D. Brito, P. V. Fenema, and S. Vermaesen, "Managing information cycles for intra-organisational coordination of humanitarian logistics", *International Journal of Services Technology and Management*, vol. 12, no. 4, p. 362, 2009.

[10] J. Holguín-Veras, N. Pérez, M. Jaller, L. N. Van Wassenhove, and F. Aros-Vera, "On the appropriate objective function for post-disaster humanitarian logistics models", *Journal of Operations Management*, vol. 31, no. 5, pp. 262–280, 2013.

[11] N. Giedelmann-L, W. J. Guerrero, and E. L. Solano-Charris, "System dynamics approach for food inventory policy assessment in a humanitarian supply chain", *International Journal of Disaster Risk Reduction*, vol. 81, p. 103 286, 2022.

[12] B. Balcik and B. M. Beamon, "Facility location in humanitarian relief", *International Journal of Logistics Research and Applications*, vol. 11, no. 2, pp. 101–121, 2008.

[13] C. L. Gilbert, L. Christiaensen, and J. Kaminski, "Food price seasonality in africa: Measurement and extent", *Food Policy*, Agriculture in Africa – Telling Myths from Facts, vol. 67, pp. 119–132, 2017.

[14] K. Peters, S. Silva, T. S. Wolter, *et al.*, "UN world food programme: Toward zero hunger with analytics", *INFORMS Journal on Applied Analytics*, vol. 52, no. 1, pp. 8–26, 2022.

[15] F. Petropoulos, D. Apiletti, V. Assimakopoulos, *et al.*, "Forecasting: Theory and practice", *International Journal of Forecasting*, vol. 38, no. 3, pp. 705–871, 2022.

[16] J. Herteux, C. Raeth, G. Martini, *et al.*, "Forecasting trends in food security with real time data", *Communications Earth & Environment*, vol. 5, no. 1, pp. 1–13, 2024, Publisher: Nature Publishing Group.

[17] D. Fuqua and S. Hespeler, "Commodity demand forecasting using modulated rank reduction for humanitarian logistics planning", *Expert Systems with Applications*, vol. 206, p. 117 753, 2022.

[18] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions", *International Journal of Forecasting*, vol. 38, no. 4, pp. 1346–1364, 2022.

[19] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015, 709 pp.

[20] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, *Temporal fusion transformers for interpretable multi-horizon time series forecasting*, 2020. arXiv: 1912.09363[stat].

[21] G. Ke, Q. Meng, T. Finley, *et al.*, "LightGBM: A highly efficient gradient boosting decision tree", in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[22] P. Baecke, S. De Baets, and K. Vanderheyden, "Investigating the added value of integrating human judgement into statistical demand forecasting systems", *International Journal of Production Economics*, vol. 191, pp. 85–96, 2017.

[23] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions", in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[24] S. Makridakis, E. Spiliotis, V. Assimakopoulos, *et al.*, "The M5 uncertainty competition: Results, findings and conclusions", *International Journal of Forecasting*, vol. 38, no. 4, pp. 1365–1385, 2022.

[25] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research", *Journal of Management Information Systems*, vol. 24, no. 3, pp. 45–77, 2007.

[26] Inter-Agency Standing Committee (IASC), "Humanitarian system-wide scale-up activation: Protocol 1: Definition and procedures", Inter-Agency Standing Committee (IASC), 2018.

[27] E. Ouchtar, S. t'Serstevens, and A. Rahman, "Demand planning in the humanitarian sector: Unpredictable or unexplored? A case study on time-series forecasting", Kühne Logistics University (KLU) and HELP Logistics of the Kühne Foundation (CHORD), 2025.

[28] World Food Programme, "WFP management plan (2025–2027)", Rome, Italy, 2024.

[29] World Food Programme, "WFP 2025 global outlook", World Food Programme, Rome, Italy, 2024.

[30] E. van der Laan, J. van Dalen, M. Rohrmoser, and R. Simpson, "Demand forecasting and order planning for humanitarian logistics: An empirical assessment", *Journal of Operations Management*, Special Issue on Humanitarian Operations Management, vol. 45, pp. 114–122, 2016.

[31] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 3nd Edition. Melbourne, Australia: OTexts, 2021.

[32] G. Athanasopoulos, R. J. Hyndman, N. Kourentzes, and A. Panagiotelis, "Forecast reconciliation: A review", *International Journal of Forecasting*, vol. 40, no. 2, pp. 430–456, 2024.

[33] D. A. Dickey and W. A. Fuller, "Distribution of the estimators for autoregressive time series with a unit root", *Journal of the American Statistical Association*, vol. 74, no. 366, p. 427, 1979.

[34] A. A. Syntetos, B. J E, and J. D. Croston, "On the categorization of demand patterns", *Journal of the Operational Research Society*, vol. 56, no. 5, pp. 495–503, 2005, Publisher: Taylor & Francis _eprint: https://doi.org/10.1057/palgrave.jors.2601841.

[35] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, Publisher: Institute of Mathematical Statistics.

[36] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning* (Springer Series in Statistics). New York, NY: Springer, 2009.

[37] L. Shapley, "7. a value for n-person games. contributions to the theory of games II (1953) 307-317.", in *Classics in Game Theory*, H. W. Kuhn, Ed., Princeton University Press, 2020, pp. 69–79.

[38] C. Molnar, *Interpretable Machine Learning*. Lulu.com, 2020, 320 pp.

[39] C. C. Holt, "Forecasting seasonals and trends by exponentially weighted moving averages", *International Journal of Forecasting*, vol. 20, no. 1, pp. 5–10, 2004.

[40] P. R. Winters, "Forecasting sales by exponentially weighted moving averages", *Management Science*, vol. 6, no. 3, pp. 324–342, 1960, Publisher: INFORMS.

[41] R. J. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for r", *Journal of Statistical Software*, vol. 27, no. 3, 2008.

[42] A. Jadon, A. Patil, and S. Jadon, *A comprehensive survey of regression based loss functions for time series forecasting*, 2022. arXiv: 2211.02989[cs].

[43] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy", *International Journal of Forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[44] H. van der Voet, "Comparing the predictive accuracy of models using a simple randomization test", *Chemometrics and Intelligent Laboratory Systems*, vol. 25, no. 2, pp. 313–323, 1994.

[45] International Monetary Fund (IMF), *World Economic Outlook, April 2025: A Critical Juncture amid Policy Shifts*. Washington, D.C.: International Monetary Fund, 2025.

[46] A. März and T. Kneib, *Distributional gradient boosting machines*, 2022. arXiv: 2204.00778[stat].

[47] E. Ogasawara, L. C. Martinez, D. de Oliveira, G. Zimbrão, G. L. Pappa, and M. Mattoso, "Adaptive normalization: A novel data normalization approach for non-stationary time series", in *The 2010 International Joint Conference on Neural Networks (IJCNN)*, ISSN: 2161-4407, 2010, pp. 1–8.

[48] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, *Optuna: A next-generation hyperparameter optimization framework*, 2019. arXiv: 1907.10902[cs].

Appendix A: Data

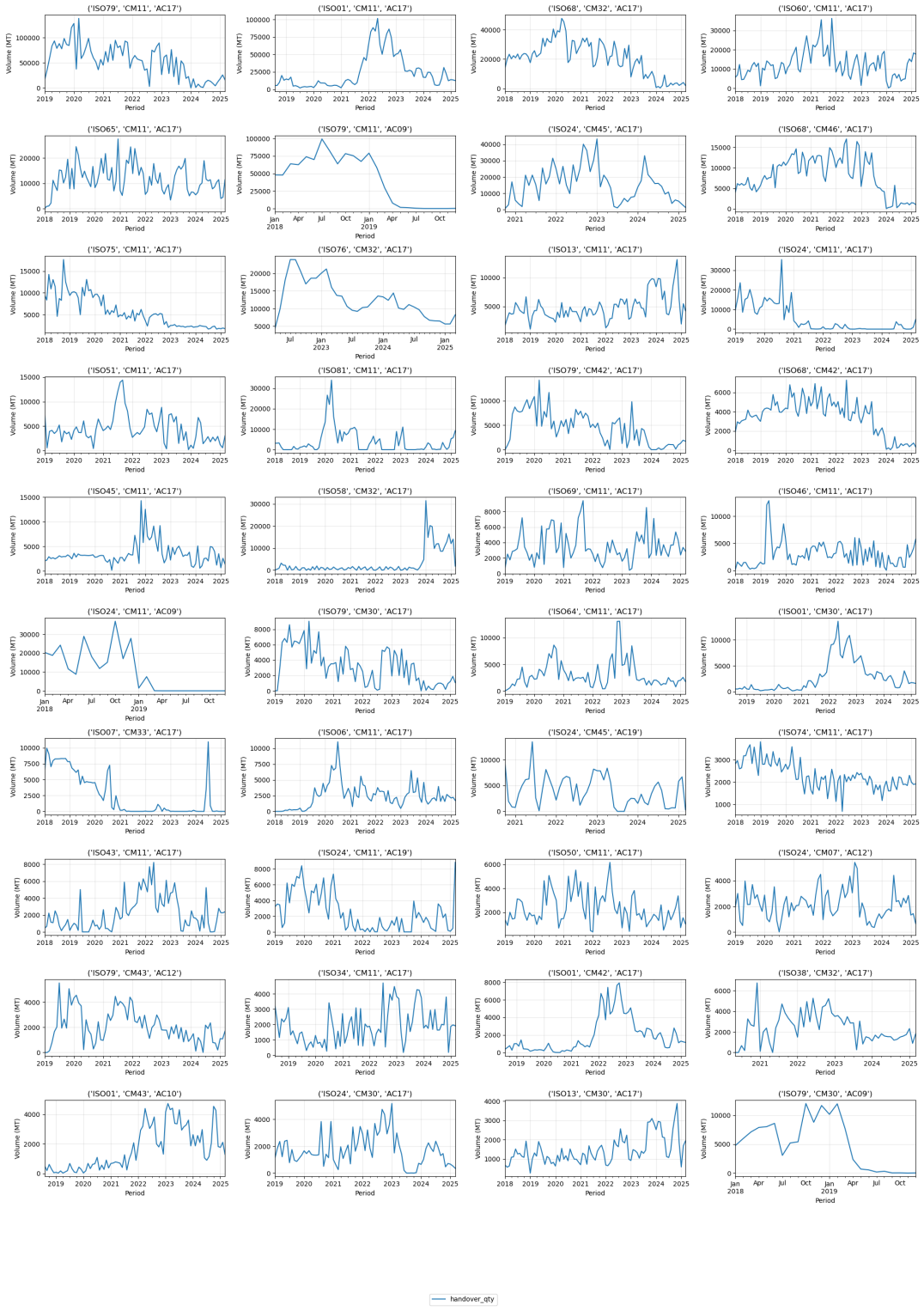


Figure 26: 40 largest segments by total demand, this is included to give a overview of the data, and shows the different profiles of the data.

Appendix B: Features for LightGBM Models

Table 8: Feature Engineering Details for Direct Model Features

Feature Name	Description
encoded_country	Integer value representing a unique country.
encoded_commodity	Integer value representing a unique commodity.
encoded_activity	Integer value representing a unique activity.
time_index	Integer representing the number of months from the start of the historical data, providing a chronological sequence.
lagN	Handover quantity from N periods prior (where N = current forecast_distance + 1).
lag12	Normalized handover quantity from 12 periods prior.
lagN_mavg_3	3-month rolling mean of 'lagN'.
lagN_mavg_6	6-month rolling mean of 'lagN'.
lagN_mavg_12	12-month rolling mean of 'lagN'.
com_lagN_mavg_12	Commodity 12-month rolling mean.
country_lagN_mavg_12	Country 12-month rolling mean.
act_lagN_mavg_12	Activity 12-month rolling mean.
IP_req	Implementation pipeline for the target period.
past_pip_bias_ratio	Historical ratio of actual handovers to pipeline requirements.
norm_factor	Normalization factor (see Equation (21)). (only for normalized model)
lagN_mavg_12	12-month rolling mean of 'lagN'.
inventory_resourced_all_periods	Aggregated resourced inventory across all relevant periods.
resourced_ratio	Ratio of current inventory usage per total inventory capacity.
PCPIPCH	Consumer Price Index Percentage Change (year-over-year inflation) [45].
NGDP_RPCH	Nominal Gross Domestic Product - Real Purchasing Power Parity (PPP) Conversion Factor available for each year and country, forecasts for 2025 also included [45].
IPC features	Estimated number of people in a specific IPC [6] index 1-5 (data is unavailable for much of training and inference).
Annual global level HO features and forecasts	'contributions', 'expenditure', 'avg_food_prices', 'food_index', 'percent_food', 'approx_expenditure_food_USD_B', 'US_inflation_rate', 'Index_on_other_costs', 'food_expenditures_volume_index'

Table 9: Feature Engineering Details for Recursive Model Numerical Features

Feature Name	Description
lag_1	Recursively updated lag_1.
lag_12	Recursively updated lag_12.
lag1_mavg_3	Recursively updated moving average 3.
lag1_mavg_6	Recursively updated moving average 6.
lag1_mavg_12	Recursively updated moving average 12.
forecast_distance	The current forecast horizon being predicted.

Table 10: Feature Engineering Details for Probabilistic Model Features

Feature Name	Description
norm_lagN_q10_6	10th quantile of the normalized 'lagN' feature over a 6-month rolling window.
norm_lagN_q25_6	25th quantile of the normalized 'lagN' feature over a 6-month rolling window.
norm_lagN_q50_6	50th quantile (median) of the normalized 'lagN' feature over a 6-month rolling window.
norm_lagN_q75_6	75th quantile of the normalized 'lagN' feature over a 6-month rolling window.
norm_lagN_q90_6	90th quantile of the normalized 'lagN' feature over a 6-month rolling window.
norm_lagN_q10_24	10th quantile of the normalized 'lagN' feature over a 24-month rolling window.
norm_lagN_q25_24	25th quantile of the normalized 'lagN' feature over a 24-month rolling window.
norm_lagN_q50_24	50th quantile (median) of the normalized 'lagN' feature over a 24-month rolling window.
norm_lagN_q75_24	75th quantile of the normalized 'lagN' feature over a 24-month rolling window.
norm_lagN_q90_24	90th quantile of the normalized 'lagN' feature over a 24-month rolling window.
norm_lagN_iqr_24	Interquartile range (IQR) of the normalized 'lagN' feature over a 24-month rolling window.

Appendix C: Hyperparameter Optimization for LightGBM Models

This appendix details the search space used for hyperparameter optimization of the LightGBM models using Optuna [48].

Table 11: Hyperparameter search space for the first models runs, optimized on only the validation set.

Hyperparameter	Optuna Suggestion Type	Search Range
n_estimators	`suggest_int`	[100, 700]
learning_rate	`suggest_loguniform`	[0.001, 0.1]
num_leaves	`suggest_int`	[20, 50]
max_depth	`suggest_int`	[3, 10]
min_data_in_leaf	`suggest_int`	[10, 50]
feature_fraction	`suggest_uniform`	[0.7, 1.0]
lambda_l1	`suggest_loguniform`	[0.001, 20]
lambda_l2	`suggest_loguniform`	[0.001, 20]

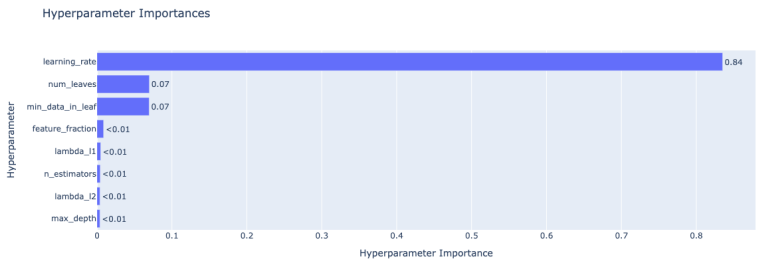


Figure 27: LightGBM direct normalized hyperparameter tuning validation parameter importance's.

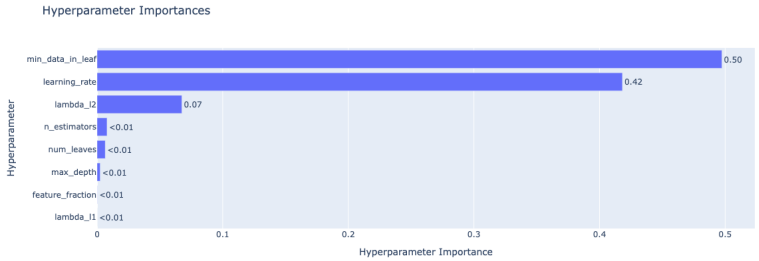


Figure 28: LightGBM recursive normalized hyperparameter tuning validation parameter importance's.

Figure 29: LightGBM direct normalized hyperparameter tuning validation trials.

Figure 30: LightGBM recursive normalized hyperparameter tuning validation trials.