A Work Project, presented as a part of the requirements for the Award of a Master's degree in Business Analytics from the Nova School of Business and Economics.

UNRAVELING THE FUTURE: A HYBRID APPROACH TO TIME SERIES FORECASTING FOR CUSTOMER SUPPORT OPTIMIZATION AT NOS

TOM TASSILO GÖRING

Work project carried out under the supervision of:

Susana Lavado

Gustavo Santos Pereira

17-12-2024

Acknowledgment: I would like to express my sincere gratitude to my advisors, Susana Lavado and Gustavo Santos Pereira, for their unwavering support and guidance throughout this work project. I also sincerely thank my Project-Based Learning group, for establishing the foundation upon which this project was built. Lastly, I am grateful to NOS and its team for providing this opportunity and their continued assistance throughout the process.

Abstract: This work project compares individual and hybrid forecasting approaches for efficient call center queue management at NOS. Besides simple individual models like theta and more complex ones such as SVMs, hybridization methods, including linear parallel hybrid and series hybrid combinations were tested. Statistical models, notably the theta model, effectively capture key patterns and outperform more complex approaches. Meanwhile, a parallel hybrid of Pareto-efficient models marginally improved performance was offset by increased complexity. Overall, this work suggests that increasing complexity is unjustified, and statistical methods can adequately forecast the task at hand.

Keywords: Forecasting, Time Series, Hybrid Combination Modeling, Data Science

This work used infrastructure and resources funded by Fundação para a Ciência e a Tecnologia (UID/ECO/00124/2013, UID/ECO/00124/2019 and Social Sciences DataLab, Project 22209), POR Lisboa (LISBOA-01-0145-FEDER-007722 and Social Sciences DataLab, Project 22209) and POR Norte (Social Sciences DataLab, Project 22209) and NOS Comunicações, S.A.

Table of Contents

1 Introduction

Forecasting is pivotal in optimizing business operations, particularly in dynamic environments like call centers, where accurate demand predictions are essential for effective resource allocation (Aldor-Noiman, Feigin, and Mandelbaum 2009). By forecasting customer needs, companies can balance staffing costs, service quality, and waiting time, directly influencing financial performance and customer satisfaction. However, achieving this balance is challenging due to inherent uncertainty in customer demand. Most notably, capacity costs, particularly labor, constitute 60%–70% of operating expenses in most call centers (Gans, Koole, and Mandelbaum 2003). The challenge of forecasting call center arrivals arises from the dynamic nature of arrival rates. Call volumes often exhibit intraday, weekly, monthly, and yearly seasonal patterns and variations alongside complex dependencies (Ibrahim et al. 2016).

Over decades, time series forecasting has been a cornerstone in scientific and industrial applications, progressing from traditional statistical methods to advanced neural network approaches. Recently, hybrid forecasting methods have gained prominence by leveraging the strengths of statistical models and machine learning techniques (Sina et al., 2023). This shift reflects an increasing reliance on model combination strategies to overcome the inherent limitations of single-model approaches. Early influential studies such as the ones by Bates and Granger (1969) and Clemen (1989) laid the groundwork for model hybridization techniques, demonstrating their superiority over single-model approaches. More recently, the M Competition series, a renowned series of forecasting challenges aimed at advancing time series forecasting methods on industrial applications, has further reinforced the effectiveness of hybrid methods (Makridakis, Spiliotis, and Assimakopoulos 2020; 2022). The benefits of model combination include enhanced forecasting accuracy through comprehensive pattern detection and modeling, mitigation of risks associated with relying on an unsuitable model by incorporating multiple forecasts, and simpler model selection by integrating different model

components (Hajirahimi and Khashei 2023). This suggests that hybrid models could be well-suited for optimizing call center operations addressing the complexity and variability of customer service data.

In the context of NOS, the complexity and variability of customer service patterns demands precise and robust forecasting solutions. To address these challenges, NOS has implemented an automated decision pipeline to efficiently route customers to appropriate call center teams, including the Standard Team (ST) and the Detractor Squad (DS). While the ST handles general inquiries, the DS is reserved for managing urgent and complex issues, particularly those with a high likelihood of recurrence. Given the DS's specialized nature and higher operational costs, optimizing this pipeline requires accurate forecasting models to ensure that only suitable customers are redirected to the DS, balancing efficiency with availability. Initial modeling efforts were made by a group of students, including myself, during Project-Based Learning (PBL). This project explored a range of statistical and machine learning models tailored for time series forecasting, providing valuable insights and a foundation for further exploration. The presented work project (WP) builds upon these findings by applying hybrid forecasting techniques to more recent data. The results and process of the PBL course are detailed in Chapter 3, *Context – Project-Based Learning*.

The objective for this WP, agreed upon with NOS, is to optimize the target model for both accuracy and stability of its predictions, progressively advancing from simpler to more complex approaches. With this challenge in mind, the following research question was developed: Can hybrid combination models improve predictions to effectively redirect potentially relapsing customers to expert call center teams?

1.1 Core Guiding Principles

Given the multitude of options available for improving a model, it is essential to establish core principles that form the foundation of the modeling strategy. Doing so ensures clarity in the

rationale behind the decisions made throughout the subsequent sections. The guiding principles are based on Armstrong's forecasting principles. These guiding principles are as follows:

1. **Building complexity progressively**: Adopting a structured approach, beginning with simpler models and methodologies and gradually incorporating more complexity as necessary. This principle ensures that the solution remains robust and aligned with the problem's requirements. This follows principles 6.6: "Select simple methods unless empirical evidence calls for a more complex approach." 7.1" Keep forecasting methods simple.", and 7.3 "Be conservative in situations of high uncertainty or instability" by Armstrong (2001, p. 693 ff).
2. **Improving upon PBL findings**: Building on insights from PBL, the focus is to enhance existing models and methodologies by building upon prior findings. This closely follows the evidence-based guideline 6.8: "Compare track records of various forecasting methods" by Armstrong (2001, p. 693 f).

Applying guiding principles was inspired by the fourth place in the M4 competition Jaganathan and Prakash (2020).

2 Literature Review

Time series analysis is a cornerstone of forecasting, focusing on understanding and modeling data that evolves over time. Key concepts are *temporal dependency* and *stationarity* (Petropoulos et al. 2022). *Temporal dependency* is a defining characteristic of time series data, representing the relationships between observations at different points in time. These dependencies are typically quantified through autocorrelation and partial autocorrelation, which capture the influence of past values on current observations. The autocorrelation function (ACF), as introduced by Yule (1927), measures the correlation between a time series and its past values over successive time intervals. The ACF provides insight into the relation of current observations to those at previous time points, thereby identifying patterns such as seasonality

or trends. Complementing the ACF is the partial autocorrelation function (PACF), developed as part of the foundational work by Box and colleagues in 1970 (2016, 5th edition). The PACF isolates the direct correlation between a time series and its past observations while controlling for the influence of intermediate observations. Meanwhile, Kwiatkowski et al. (1992) define *stationarity* in the context of time series analysis as a property where the statistical characteristics of the series, such as mean and variance, remain constant over time. This implies that the time series does not exhibit trends or seasonal effects, and its behavior is consistent throughout its duration. The authors introduce the Kwiatkowski–Phillips–Schmidt–Shin (KPSS**)** test to assess the null hypothesis that a time series is stationary against the alternative hypothesis of the presence of a unit root, which would indicate non-stationarity. Similarly, the Dickey-Fuller test is another widely used method for determining the presence of a unit root in a time series, assessing whether the time series exhibits non-stationarity (Dickey and Fuller 1979).

2.1 Linear Models

Linear, statistical time series models are simple and interpretable tools in statistical analysis based on theoretical assumptions about the data-generation process. They are easy to implement and yield robust results if assumptions like stationarity and linearity hold. However, violations of these assumptions can lead to biased and unreliable forecasts, making it crucial to ensure their validity for accurate application and results. To enhance evaluation and ensure robustness, Petropoulos et al. (2022) suggest that new models should always be compared to, at minimum, one naïve model like random walk and one popular general forecasting method like Autoregressive Integrated Moving Average (ARIMA). The following sections will discuss models applied during the optimization process.

The naïve model assumes that the most recent observation is the best predictor for future values, projecting the data forward without adjustment. The average method is more sophisticated since

predicted values are based on the mean of past observations, but it still assumes no underlying trend or seasonality in the data. Past literature has shown that such models perform well regarding call center data, especially in a forecast timeframe of more than 3 days, even described as difficult to beat. (Taylor 2008; Aldor-Noiman, Feigin, and Mandelbaum 2009). Regarding popular general forecasting models, three were chosen: Seasonal Autoregressive Integrated Moving Average (SARIMA), theta, and Exponential Smoothing (ES). The SARIMA model extends the ARIMA framework by incorporating seasonal components to address recurring patterns in data. While ARIMA captures short-term and non-seasonal dependencies through autoregression (AR), differencing (I), and moving average (MA) parameters, SARIMA introduces seasonal versions of each parameter (S), making it ideal for time series with distinct periodicity, such as daily, weekly, or monthly call volume fluctuations (Box et al. 2016). The theta model is a decomposition-based method that applies a linear trend component and a smoothed seasonal component to forecast time series data (Assimakopoulos and Nikolopoulos, 2000). Renowned for its simplicity and accuracy, the theta model was highly effective in the M3 Competition and has since become a reliable tool for series with clear seasonal patterns and minimal complexity (Makridakis and Hibon 2000). ES is a forecasting method introduced by Brown (1959) that applies exponentially decreasing weights to past observations, giving higher importance to recent data. Holt (2004), originally published in 1957, expanded simple ES to account for trends by adding a trend component, enabling it to forecast series with steady increases or decreases. For more complex series with trends and seasonality, Winters (1960) incorporates seasonal components, making it ideal for series with recurring patterns like daily or monthly fluctuations.

2.2 Non-linear Models

Non-linear models are vital in time series forecasting because they capture complex patterns without linear assumptions, making them ideal when the data-generating process is unknown

or non-linear. Although these models often achieve high predictive accuracy, their complexity and reduced interpretability necessitate careful evaluation. Support Vector Machines (SVMs) and Gradient Boosting Machines (GBMs) were applied in this WP.

SVMs, introduced by Boser, Guyon, and Vapnik (1992), are supervised learning methods for classification and regression, often referred to as Support Vector Regression when deployed for continuous predictions. SVMs identify an optimal hyperplane to minimize prediction errors within a tolerance margin and employ kernel functions to reflect relationships. SVMs have demonstrated potential in time series prediction; Nie et al. (2012) showed that combining SVMs with methods like ARIMA improves forecasting accuracy by effectively capturing linear and non-linear patterns. GBMs are machine learning methods that build predictive models by sequentially combining weak learners, typically decision trees, to minimize errors. They apply gradient descent to optimize an objective function, improving performance iteratively (Friedman 2001). GBMs can excel in time series forecasting by leveraging correlations across related series and employing different configurations to enhance robustness and accuracy. Configurations include custom loss functions, recursive and non-recursive approaches, and advanced feature engineering such as calendar effects and lagged variables. Combining forecasts from different configurations often results in exceptional forecasting accuracy (Makridakis, Spiliotis, and Assimakopoulos 2022). XGBoost, applied in PBL, is a widely used GBM known for its scalability. It utilizes techniques like regularization, parallel processing, and flexible tree-pruning, making it well-suited for structured data and large-scale datasets (Chen and Guestrin 2016). LightGBM is a gradient-boosting framework like XGBoost that instead prioritizes speed and memory efficiency. Using histogram-based decision trees and leaf-wise tree growth, it accelerates computation without compromising accuracy. Ideal for large datasets and high-dimensional problems, LightGBM is effective for complex machine learning ML tasks. In time series forecasting, it offers comparable benefits to XGBoost but often

achieves faster performance with similar accuracy, making it preferred for time-sensitive applications (Ke et al. 2017). It gained significant recognition for its superior performance in the M5 competition (Makridakis, Spiliotis, and Assimakopoulos 2022).

2.3 Hybridization

Recent advancements in time series forecasting highlight the limitations of using pure statistical or machine learning models independently. The M4 and M5 Competitions revealed that pure models performed poorly compared to hybrid approaches and that integrating features from statistical and machine learning domains often leads to superior performance (Makridakis, Spiliotis, and Assimakopoulos 2020; 2022). Numerous studies support that hybrid models outperform individuals models, achieving greater accuracy and robustness (see, for instance, Armstrong 2001; Bates and Granger 1969 ; Makridakis and Hibon 2000).

While Ensemble methods involve aggregating multiple models of the same type to improve accuracy and robustness, hybrid methods combine different types of models or algorithms to leverage their complementary strengths (Ardabili, Mosavi, and Várkonyi-Kóczy 2020). In this WP, since different types of models are combined, we refer to them as hybrid models. However, it is important to note that this terminology is not used uniformly across the literature.

According to Hajirahimi and Khashei (2023), hybrid models can be classified into four categories: (1) Preprocessing-based hybrid models, (2) parameter optimization-based hybrid models, (3) components combination-based hybrid models, and (4) postprocessing-based hybrid models. Preprocessing-based hybrid models in time series forecasting combine data preprocessing techniques, such as decomposition or denoising, with forecasting methods to improve model accuracy and robustness by isolating key patterns or removing noise. Parameter optimization-based hybrid models leverage optimization algorithms like Bayesian optimization to fine-tune model parameters by optimizing for a given error term (Sina et al. 2023). However, these models can be computationally intensive and may involve complex optimization

processes, which increases the time and resources required for model development. Postprocessing-based hybrid models enhance forecasting accuracy by processing residuals from initial model forecasts using techniques like Kalman filters. This approach captures remaining patterns in the data for more complete modeling but can increase computational time and implementation costs due to the additional processing steps involved. In contrast, components combination-based hybrid models focus on combining outputs of individual forecasting models, leveraging their unique strengths to enhance overall accuracy. While this method can effectively capture both linear and nonlinear patterns in the data, it may also increase model complexity and computational requirements (Hajirahimi and Khashei 2023). In this WP, components combination-based hybrid models were chosen due to their ability to capture both linear and nonlinear patterns, which was considered a decisive advantage in the use case at hand.

2.3.1 Combination-based Hybrid Models

Combination-based forecasting methods, which integrate multiple models to improve accuracy, have been highly effective in practical applications early on (Armstrong 2001; Makridakis and Winkler 1983; Clemen 1989). More recently, the M4 competition highlighted that combinations of statistical and ML models tend to outperform models relying on individual approaches. This is attributed to the ability of combined methods to capture different components of data patterns, effectively averaging out individual model errors for greater accuracy (Makridakis, Spiliotis, & Assimakopoulos, 2020, p. 64). Building on this, the M5 Competition highlighted the strong performance of combinations of different configurations of GBMs (Makridakis, Spiliotis, and Assimakopoulos, 2022).

Hajirahimi and Khashei (2019) classify combination-based hybrid models into three main structures. The parallel structure runs multiple models simultaneously and combines their outputs. The series structure uses the output of one model as the input to another in sequence.

The parallel–series structure combines elements of both parallel and series configurations to leverage their advantages.

2.3.1.1 Combination-based Parallel Hybrid Models

By combining multiple models, parallel hybrid models enhance robustness and reliability, leading to more accurate forecasts—especially for complex or nonlinear time series data where single models may fail to capture all underlying patterns (Hajirahimi and Khashei 2019). One of the most popular approaches is linear combination methods employing static weighting schemes, which apply fixed weights (such as simple averaging and minimum error methods) to individual models. In contrast, dynamic-based parallel hybrid models adjust weights in real time based on data or model performance, achieving greater accuracy through enhanced adaptability. While linear hybrid approaches are prevalent, a review exploring combination approaches has shown that nonlinear combined models yield better results by capturing complex relationships that linear methods might overlook (Hajirahimi and Khashei 2019). However, due to their increasing complexity, nonlinear combination approaches fall outside the scope of this WP.

The three most mentioned linear weighting approaches in literature are the simple average, minimum error weighting method, and the Variance-Covariance (VarCov) method. While the simple average method is self-explanatory, the minimum error weighting methods aim to minimize an error term, with mathematical techniques like Ordinary Least Squares (OLS) often outperforming metaheuristic algorithms due to their superior performance and lower computational cost. The VarCov method proposed by Bates and Granger (1969), calculates weights to minimize the combined forecast error variance by using past forecasting performance to compute individual error variances and covariances.

2.3.1.2 Combination-based series hybrid models

The most common sequence in series hybrid models, where one model is trying to predict the error of another model, is linear-nonlinear, used in over 90% of studies reviewed by Hajirahimi and Khashei (2019). Comparative studies of linear-nonlinear versus nonlinear-linear sequences are limited but suggest that nonlinear-linear models may outperform their counterparts in certain contexts. Additionally, the research by Hajirahimi and Khashei (2019) highlights that the accuracy of series hybrid models improves when more accurate individual models are selected as components. A notable example is the ARIMA-SVM hybrid model. SVMs are often chosen in the second stage due to their effectiveness in nonlinear regression. Studies have shown that ARIMA-SVM models generally achieve more accurate results than ARIMA-Artificial Neural Network hybrids (Hajirahimi and Khashei 2019).

2.3.1.3 Combination-based parallel-series hybrid models

Researchers have developed parallel–series hybrid models to combine the strengths of series and parallel structures, aiming to improve forecasting accuracy. Studies indicate that these models can outperform purely series-based approaches. However, there is no consensus on their superiority over series or parallel models for all forecasting tasks, as their effectiveness varies by context (Hajirahimi and Khashei 2019). The design and implementation of parallel–series models are complex and computationally intensive, potentially limiting their practicality. Furthermore, research on these models is limited, with findings based on few comparative studies and specific cases (Hajirahimi and Khashei 2019)For these reasons, this architecture will not be investigated further in this work, following the core guidelines.

3 Context – Project Based Learning

The foundation for this forecasting approach was set in the Master's in Business Analytics PBL program, which addressed the same problem at NOS. PBL is an initiative within the Master's in Business Analytics designed to enable students to work with companies on real-world

challenges these organizations face. The preparatory work performed in PBL can be divided into the following: describing the data, understanding the problem, understanding the data, developing the models, planning for testing, and determining the final pipeline and library. This structure was given by the PBL course.

The data provided by NOS for the PBL project contained 11 tables with approximately 450 variables. The tables were derived from different sources and were originally stored in NOS's Data Lake. They contained data related to customer calls, the equipment, and services provided to customers currently and in the past, house visits scheduled or performed by technicians, and calls' locations. To protect the privacy and confidentiality of individuals, the data was pseudonymized. This led to the exclusion of tables about locations, as they could not be linked with the remaining data. The data timeframe was from May 31, 2023, to September 29, 2023. While sharing the same structure, the data provided for PBL differed from the data used in this WP and, therefore, will not be further detailed.

Outlining the problem, the main stakeholders were identified as the NOS, its call center, its clients, and NOVA School of Business and Economics (NOVA SBE). Positioning the project within the business context clarified its impact on these stakeholders. The call center benefits from improving resource allocation, reducing employee strain, and enabling more effective achievement of daily targets. Clients experience quicker issue resolution, enhancing their satisfaction and loyalty to NOS. NOVA SBE and its students gain valuable hands-on experience and strengthen their partnership with NOS, highlighting the effectiveness of data-driven solutions in tackling business challenges. It was also clarified that the approach is implemented exclusively for non-business clients.

The critical objective of PBL was defined as "Creating an accurate model for flagging clients to be redirected to the specialized call center team of the technical line.", with the assumptions of data accessibility and accuracy. Key potential constraints included the predictive accuracy

of the first model, the limited project timeline, access to the Data Science Knowledge Centre's servers, and exclusivity to the technical line.

While the project aims to enhance NOS's efficiency and customer service, it also raises ethical considerations regarding privacy, fairness, and transparency. To address these, the data was pseudonymized, mitigating privacy concerns for the Nova SBE team. Focusing on predicting the number of customers to redirect rather than identifying specific individuals further reduces the risk of biased outcomes or inequities. NOS ensured compliance with the General Data Protection Regulation through data pseudonymization and rigorous privacy assessments. The Nova SBE team adhered to strict data privacy and security protocols, with all major decisions made collaboratively with NOS.

To efficiently redirect customers, NOS developed a two-model pipeline. The first model assigns a score to each customer indicating the likelihood of them calling within the next 15 days—defined here as "relapsing". The second model determines how many customers to flag for redirection to the DS, using an ordered list ranked by scores from the first model. Its target variable is the number of customers flagged to meet a variable goal for incoming calls to the DS, anonymized as constant *c* instead of a fixed numerical value. Figure 3 in the appendix illustrates a simplified pipeline example for a single day.

3.1 Model Evaluation

To evaluate the models, Mean Average Percentage Error (MAPE) and Smoothness Index (SI) were applied. MAPE is calculated as the mean of absolute percentage differences between forecasts and observed values. It was chosen for its interpretability, providing a clear and straightforward metric of the model's accuracy. Also, unlike squared errors, MAPE does not emphasize larger errors. The SI measures prediction variability, with lower values indicating smoother, more consistent changes and higher values reflecting greater variability. Given the

high importance of stable predictions to prevent operational disruptions in the call center, SI was chosen as a secondary metric to assess the models' applicability.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \times 100$$

Equation 1: Mean Average Percentage Error

$$SI = \frac{\sum_{t=1}^{n}(\hat{y}_{t+1} - \hat{y}_t)^2}{n \times Var(\hat{y})}$$

Equation 2: Smoothness Index

MAPE and SI were calculated using the actual call arrivals corresponding to the predictions, and the model's performance is evaluated based on Pareto-efficient hyperparameter configurations. In this context, Pareto efficiency refers to a situation where the model's performance cannot be improved in one metric without worsening its performance in the other. This approach optimized the balance between accuracy and stability. According to NOS, it is preferable to overestimate rather than underestimate the number of calls slightly. Therefore, MAPE is applied to a range instead of the exact target, allowing a buffer for slight overestimations. Predictions within the target plus a defined tolerance are treated as zero error, ensuring minor overestimations are not penalized. The metrics were computed using a Temporal cross-validation (TCV) approach. TCV is a validation technique in which the dataset is split into sequential train and test sets based on time. In this approach, training occurs on past observations, and predictions are made on future data, ensuring that the temporal order of the data is preserved to avoid data leakage. In addition to the train and test sets, an Evaluation Window (EW) was used — a period during which all model predictions are assessed to ensure comparability across models. In our configuration, the models predicted for a given test set, in this context, always one day, and then the absolute percentage error was calculated. Thereafter, the train and test set shifted by plus one day (see Figure 1), repeating the process. The percentage errors in the EW were then averaged to create the MAPE for the EW.

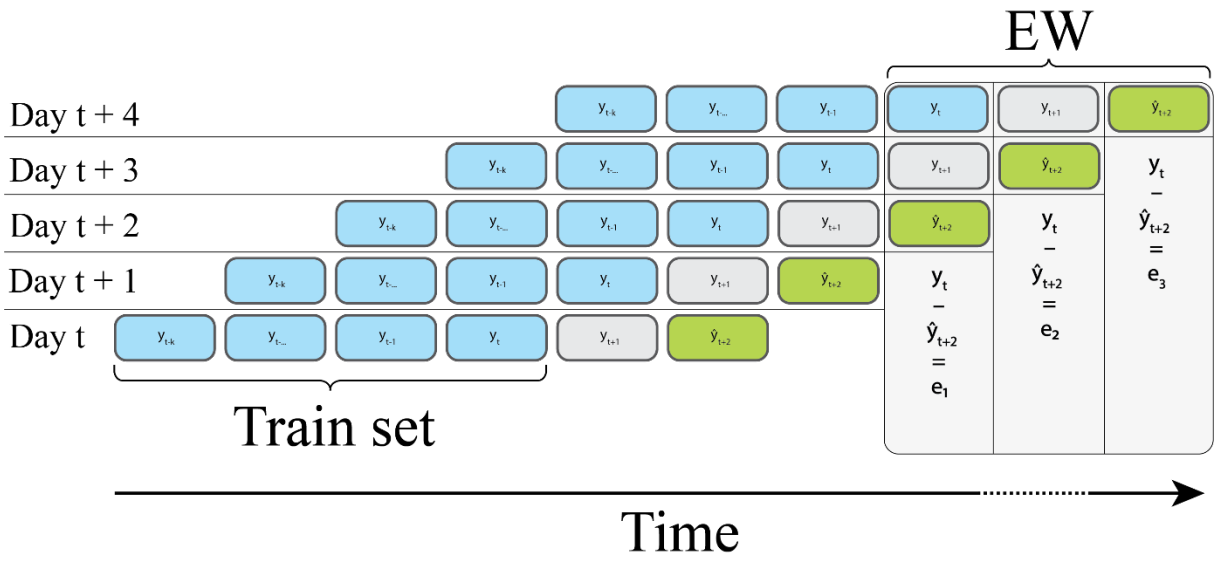

Figure 1: Illustration of TCW and EW

Figure 1 illustrates the TCV process, where y_{t-n} represents the observed value for a given day, $\hat{y}_{t+2}$ is the model's prediction for Day t + n, and e_n denotes the error for each day in the EW. The time t is always interpreted in the context of its respective day. The train set is highlighted in blue, while the target day is marked in green for each day. Due to operational constraints, a one-day gap was required between the train and test sets, shown in grey as y_{t+1}. Consequently, the model learned the relationship between the training set (train set) and the test set on the second subsequent day, y_{t+2}, skipping the y_{t+1}. This process emulated the deployment of the model within NOS. The training sets could differ between models, with ranges from 5 to 40 days in increments of 5 days, with weekends excluded, as the DS operates only on business days. Thus, each 5-day increment corresponds to one business week. In the case of the PBL project, the EW comprised the last 40 days of the data, excluding one outlier. This outlier was omitted as it may be linked to operational disturbances, according to NOS.

3.2 Modeling

The modeling approach applied use-case-specific statistical and ML models optimized via a custom grid search. The search tuned hyperparameters, enabling the selection of the best-performing models. To reduce the impact of outliers, we applied the interquartile range (IQR)

multiplier with a factor of 1.5, adjusting outlier values by adding 20% of the difference between the original value and IQR multiplied by 1.5. This preserved some outlier influence while preventing destabilization.

As baselines within PBL, three options were implemented: The average of the days within the training set, the average of the days which are the same weekdays as the day to predict within the training set, and the model that is in place at NOS, an ARIMA (ARIMA NOS). The training set for the baseline models is always 30. The Average model provides the simplest benchmark, allowing a basic comparison of more complex models. The Weekday Average model captures consistent weekly patterns in the data, accounting for peaks and valleys, but is vulnerable to outliers and data drifts. Finally, ARIMA, used by NOS, struggles with sudden changes in the data but serves as a crucial benchmark for comparison.

For univariate time series forecasting ARIMA, SARIMA, ES, and the theta model were implemented. As shown in Table 1, the SARIMA optimized during PBL (SARIMA PBL) model achieved the lowest MAPE of the statistical models, demonstrating strong accuracy and stability in predictions. The SARIMA PBL model used a training set of 20 days and included both non-seasonal and seasonal components. The non-seasonal parameters ($p = 2$, $d = 0$, $q = 1$) allowed the model to use past values of the series without differencing to predict future values, while the seasonal parameters ($P = 0$, $D = 0$, $Q = 1$, $m = 20$) indicated that the seasonal cycle repeated monthly rather than weekly, as initially expected. When comparing SARIMA PBL to the Benchmark ARIMA NOS, the two models showed similar non-seasonal components. The NOS ARIMA model indicated slightly less accuracy but better stability. None of the models used differencing, showing that treating the data as stationary improved performance. To enhance predictive power, multivariate forecasting techniques were explored, integrating variables influencing the time series. SARIMAX, Prophet and XGBoost were applied within PBL. In comparing the performance of the multivariate models, XGBoost yielded the best

performance. Meanwhile, Prophet demonstrated superior smoothness in its predictions, but it lacked accuracy, with a higher MAPE. Despite Prophet's smoother output, the reduction in error provided by XGBoost led the team to favor it as the more effective model. In conclusion, PBL provides a deployable SARIMA solution and sets the stage for improving XGBoost. SARIMA offers a robust balance of accuracy and interpretability, while XGBoost shows promise for future improvements through further optimization.

Model	Hyperparameters	Train set size	MAPE	SI
Average	-	30	12.07%	0.06
Weekday Average	-	30	16.78%	1.55
ARIMA NOS	p = 1, d = 0, q = 1	30	13.22%	0.1
SARIMA PBL	(non-seasonal: p = 2, d = 0, q = 1) (seasonal: P = 0, D = 0, Q = 1, m = 20)	20	6,89%	1.44
XGBoost PBL	Estimators: 20, Depth: 35, L1: 0.1, Objective: reg. absolute error, learning_rate: 0.15, booster: Dart	10	6.75%	2.13

Table 1: Best performing models from PBL

4 Methods

4.1 Differences in Approach to PBL

Discussions with NOS for this WP emphasized developing an accurate model specifically tailored to the given time series data without measuring error based on calls or using a predefined tolerance zone. This approach avoids distortions from converting predictions to actual calls, as shown in Figure 4 in the appendix, and eliminates bias toward overpredictions. The rationale is that this approach simplifies model predictions' interpretability, and a well-performing model can still be trained to a slightly higher goal, allowing for controlled overprediction when necessary. To further simplify the interpretability of model performance, the SI was replaced with the Mean Absolute Percentage Change (MAPC) to measure the average absolute percentage change between consecutive data points.

$$MAPC = \frac{1}{n-1}\sum_{t=2}^{n}\left|\frac{y_t - y_{t-1}}{y_{t-1}}\right| \times 100$$

Equation 3: Mean Absolut Percentage Change

Due to the increased data and number of models, a plain grid search was computationally infeasible. Instead, the models were optimized towards both MAPE and Mean Squared Error (MSE) using Optuna, an open-source framework for automated hyperparameter optimization. Optuna uses Bayesian optimization and pruning to reduce grid search costs, offering dynamic search space design and improved model performance with minimal manual tuning (Akiba et al. 2019) Optuna operates in trials, each exploring a unique hyperparameter configuration. One hundred trials per model were applied.

The final difference from PBL was the data. The model assigning call likelihood scores was improved, altering customer rankings and overall data behavior. Additionally, more and more recent data was available. The available data covered the time frame from November 1, 2023, to November 15, 2024. Of these 380 days, weekends and holidays were excluded since the DS operates only on business days. Additionally, three outliers caused by operational disturbances were removed, clearly identifiable in Figure 5 in the appendix. This resulted in 258 observations in the final dataset, shown in Figure 6 in the appendix. The data was split into training, validation, and holdout sets. The training set, shown in Figure 7 in the appendix, comprises the first 145 observations (June 1, 2023, to May 31, 2024), used exclusively for model training. The validation set, shown in Figure 8 in the appendix, used for hyperparameter optimization consists of the following 83 observations. The holdout set, shown in Figure 9 in the appendix, includes the last 31 observations (October 1, 2024, to November 15, 2024). This holdout set was reserved for evaluating the model's generalization performance.

The Shapiro Wilk and the Anderson-Darling test indicated that the data is not normally distributed. As for stationarity, the ADF test returned a p-value of 0.025, suggesting it was present, while the KPSS test returned a p-value of 0.1, suggesting non-stationarity. These

conflicting results highlighted the need for further analysis. The decomposition of the time series revealed a weekly seasonality and changes in trend, but no consistent overall trend, as seen in Figure 10 in the appendix, which may explain the contradictory stationarity outcomes. The ACF and PACF indicate that the five previous business days are the most strongly correlated with the current day, with weaker but occasionally significant correlations extending further back, as shown in Figure 12 in the appendix, which could explain the results of a seasonality parameter of 20 in PBL's SARIMA.

4.2 Modeling

This section will explain how models were applied and the reasoning behind the choices while gradually moving from lower to higher complexity. To diversify outputs and capture different patterns in the data, as suggested by Makridakis, Spiliotis, and Assimakopoulos (2022), each model's hyperparameters were optimized for MAPE and MSE. An outlier removal technique was applied to the train set for each day, with IQR method being the chosen one, with a tested multiplier between 1 and 10. The training set size was optimized for between 5 and 140 days, and these ranges were applied across all models. As baselines, a naïve model and a SARIMA model were implemented. The naïve model predicts the value from the last available observation two days ahead, while the SARIMA model, deployed during the internship, uses the hyperparameters $p = 1$, $d = 0$, $q = 1$, $P = 1$,$D = 0$, $Q = 0$, and $s = 5$.

Further implemented statistical models were simple average, weekday average, smoothened weekday average, SARIMA, ES model, and the theta model. Multiple averages were deployed because they have been shown to perform well on call center data by Taylor (2008) and were identified as potentially suitable in the PBL work. Next to the simple average and the weekday average, a smoothed weekday average was deployed. It reduces sensitivity to outliers by applying a Savitzky-Golay filter before calculating weekday averages. This filter smooths data by fitting consecutive subsets of points with low-degree polynomials using linear least squares,

effectively preserving features like peak heights and widths that other methods might distort (Savitzky and Golay 1964). This adds two parameters for optimization: the window length and the polynomial order. Although these models achieve competitive accuracy, they cannot be deployed to production without constant monitoring due to their slow adaptability to data drifts, making them unsuitable for production but applicable for hybridization, as discussed with NOS. Moving to more complex approaches, statistical models capable of capturing seasonality were applied, namely SARIMA, ES, and theta. Each model was optimized categorically for a period of 5 or 20 days, representing weekly and monthly seasonality. These models were selected based on their proven performance in real-world challenges (Makridakis and Hibon 2000; Makridakis, Spiliotis, and Assimakopoulos 2020) and based on their application in PBL.

The most complex individual models applied were the multivariate non-linear models SVMs and LightGBM. SVMs were selected due to their proven ability to handle nonlinear relationships and sparse data well (Sapankevych and Sankar 2009). LightGBM was selected over XGBoost due to its often similar performance but much less resource-intensive application and its dominance in the M5 competition. Recursive and non-recursive approaches were implemented to diversify model behavior, following the findings from the M5 competition (Makridakis, Spiliotis, and Assimakopoulos 2022). The non-recursive approach predicts from the last given observation in the training set directly to the target two days ahead. This requires learning the relationship between the last day of the test set and the second subsequent day. The features used were daily aggregations (mean, standard deviation, and sum) of the call durations, the hold time, the number of calls to the DS, the number of calls to the technical line, and the score given to each customer by the previous model. These were complemented by cyclical encoding of days of the week, month, and year as calendar features, along with the five most recent observations as lagged features, based on significant correlations in the PACF plot. In contrast, as no aggregated features are available for the days between the last day of the training

set and the target, the recursive approach relies on calendar and lagged value features. It uses these and its prediction as input to forecast subsequent days.

Complexity was then further increased by introducing hybrid parallel combination modeling approaches. Here, the simple average has demonstrated strong performance in multiple M competitions and is the most extensively researched method (Makridakis, Spiliotis, and Assimakopoulos 2022). In addition, other well-documented approaches, namely error minimization and VarCov, were implemented (Hajirahimi and Khashei 2019). Bayesian, Ordinary Least Squares (OLS), and Non-Negative Least Squares (NNLS) were implemented among the error minimization methods. Bayesian was chosen for its versatility, OLS for its superior performance, as reported in the literature, and NNLS for its ability to improve the stability of OLS. For Bayesian weight optimization, 50 trials were conducted each day for optimizing weights with a constraint for each weight to be between zero and one before normalization. All weighting approaches were implemented dynamically, updating weights daily based on the model errors from the previous five days of the training set. The WP focused on these linear weighting approaches, excluding less common or non-linear combination methods.

To identify the best weighting approach, all optimized models, excluding baselines, were combined using each implemented weighting method, and the method yielding the best results was used for further hybridization. Next, following the findings by Makridakis, Spiliotis, and Assimakopoulos (2022), which suggest that combining the best models produces the best hybrid models, Pareto-efficient models between MAPE and MAPC were combined. Additionally, models with a MAPE below 13, the Pareto-efficient linear models, and the Pareto-efficient non-linear models were combined.

The final step was adding complexity through series hybridization. As previously, series hybrids were tested by combining the best models. These included combinations of the best

individual non-linear and linear models in both directions—starting with either the linear or non-linear model, as well as the best purely non-linear or purely linear parallel combined models. For hybrid combinations predicting the error of a previous model in the series, each individual model was first individually optimized before combining the predictions. Optimal hyperparameters and searched hyperparameter ranges can be found in Table 3 in the appendix.

5 Results

Figure 2 summarizes the results of all models, while Table 2 displays the five models with the lowest MAPE and the baselines. Baselines are marked with black circles, dotted for the naïve model and solid for SARIMA, while the two models with the lowest MAPE are marked with red circles, dotted for the Theta model optimized for MAPE and solid for the parallel hybrid model consisting of the Pareto-efficient models and a dotted red line connecting Pareto-efficient models. Additionally, individual model names are followed by the error term they were optimized for, and parallel hybrids are followed by the applied weighting method, as shown in all tables and figures from now on. Among the linear models, the theta model optimized for MAPE achieved the lowest MAPE. In this regard, the best-performing non-linear model was the recursive implementation of the SVM, while the model with the overall lowest MAPE was the parallel hybridization of the individual Pareto-efficient models; it marginally outperformed the theta model. The best models, the hybridization of the individual Pareto-efficient models and the theta model achieved MAPEs of 10.52% and 10.59%, outperforming the deployed SARIMA by nearly five percentage points on the EW while maintaining a similar MAPC. Notably, the naïve model was outperformed in MAPE by almost eight percentage points and had the worst MAPC, being the only model with a MAPC higher than the actual data. Further model performances can be found in Table 4 in the appendix, and the models contained in the different hybridizations can be seen in Table 5 in the appendix.

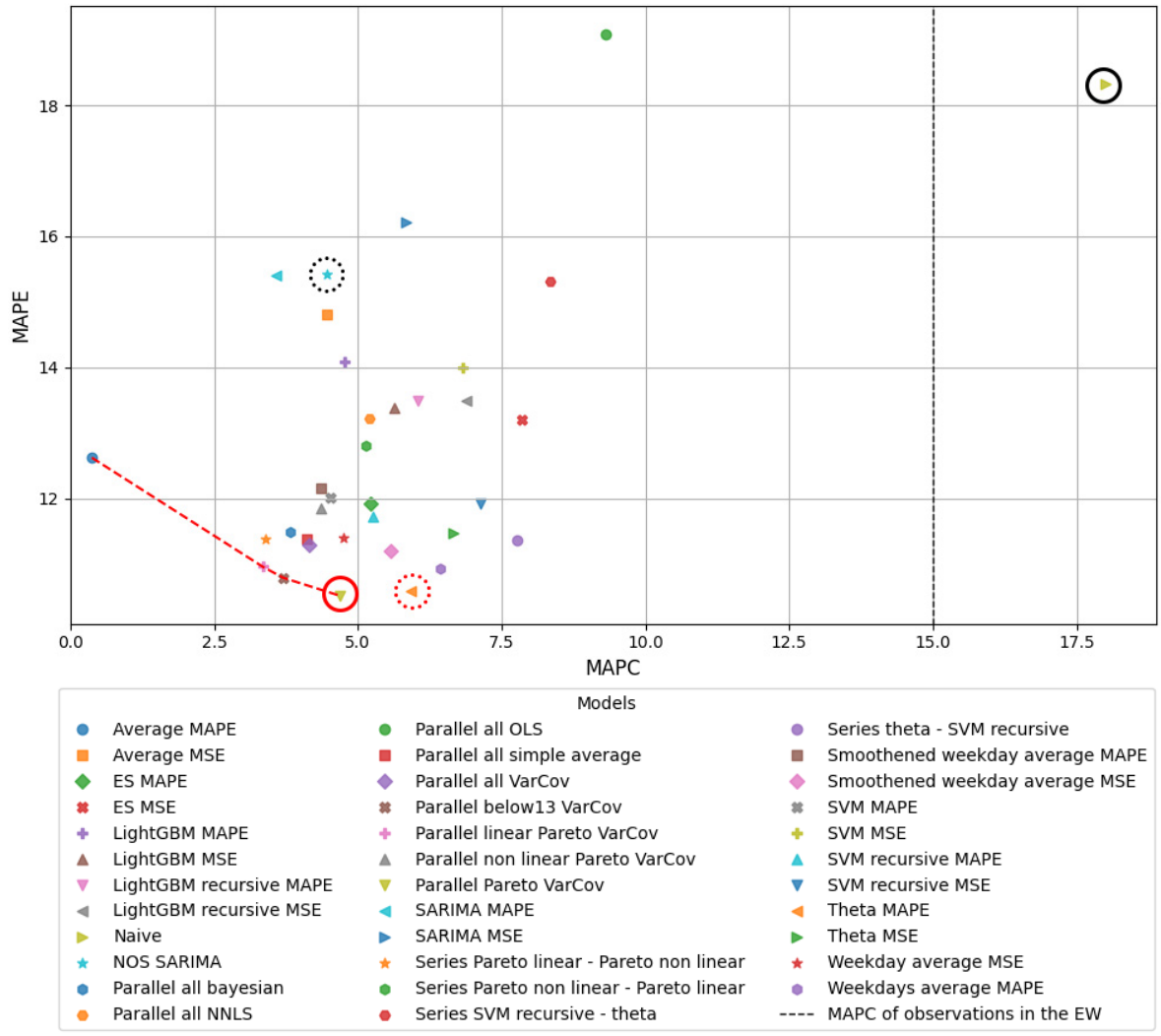

Figure 2: Evaluation of all models

Table 2: Ranges and optimal hyperparameters of all applied models

MODEL	MAPE	MSE	MAE	MAPC
Parallel Pareto VarCov	10.52	41802239	5008	4.69
Theta MAPE	10.59	47285069	5071	5.92
Parallel below 13 VarCov	10.79	44183548	5096	3.70
Weekdays average MAPE	10.93	41112555	5139	6.44
Parallel linear Pareto VarCov	10.96	43623707	5143	3.34
NOS SARIMA	15.42	85489146	7359	4.46
Naive	18.33	143282066	8850	18.01

Simpler models like theta and averages outperformed more complex non-linear models, with SVMs close behind but LightGBM performing the worst. The optimized and baseline SARIMA models lagged significantly behind other models. In the series hybrid combinations, no model could improve the predictions of the previous model; instead, they increased the error. In

contrast, parallel hybrid combinations using VarCov showed slight improvements by maintaining MAPE while reducing MAPC, leading to better stability.

6 Discussion and Conclusion

This WP focused on improving a time series models to predict the number of customers to flag for redirection to an expert call center team. Tested approaches were averages, linear statistical models, univariate non-linear models, and hybridizations of all mentioned models. While a hybrid approach achieved the lowest MAPE, the linear statistical theta model showed the best balance of simplicity and performance. These results pose the question: Is the slight improvement in stability worth the added complexity?

LightGBM excelled in the M5 competition by managing numerous features and cross-learning across multiple time series, pooling them to utilize large amounts of data and enhance performance. (Makridakis, Spiliotis, and Assimakopoulos 2022). Calendar and lagged features were available, similar to those often used in the M5 competition. Therefore, the weaker performance in this application most likely results from a different data behavior or data sparsity, due to training sets only reaching up to 140 observations. Thus, an explanation as to why SVMs outperform LightGBMs in this use case might be their capability to using kernel functions to map data into higher-dimensional spaces, capturing complex patterns while maintaining a clear separation between relevant and irrelevant observations.

Additionally, statistical models like the theta model often excel with sparse data due to their simplicity, robustness, and low reliance on large datasets, given that the assumptions, such as stationarity or the seasonality cycle made by the model, hold, as they do in this use case. The theta model also benefits from having few hyperparameters to optimize. With a validation set of 83 observations and a limit of 100 Optuna trials per model due to time constraints, simpler models like theta are quicker to tune and less prone to overfitting hyperparameters to the validation set's behavior. By decomposing time series into level, trend, and seasonality, theta

effectively captures periodic patterns and mitigates noise, even with limited observations. In this case, theta appears particularly well-suited to the data, as its design optimally captures the weekly seasonality and other data behaviors.

Although introducing a parallel hybrid approach by combining Pareto-efficient models and using the VarCov method to compute daily weights dynamically, results in a slight improvement over the theta model's MAPE and reduces the MAPC by approximately one percentage point, these gains come at the cost of substantially increased complexity. Given these trade-offs, the added complexity of hybridization does not appear justified for this particular use case. Additionally, most models had a lower MAPC than the data in the EW (14.9985%), indicating that predictions were already cautious, as seen in Figure 2.

These findings emphasize that increasing model complexity is superfluous when a simpler statistical model like theta can effectively capture the underlying patterns of a time series. The theta model delivers reliable and interpretable forecasts by focusing on fundamental components such as level, trend, and seasonality. This highlights the value of aligning model choice with the characteristics of the data, demonstrating that a simple, well-chosen statistical model can capture the time series' underlying patterns in this use case.

6.1 Limitations and Future Work

The primary limitations of this work include time constraints from the internship, data sparsity, restricted access, and implementation constraints. Data sparsity favored simpler models, restricted access limited further feature exploration or exploitation of cross-learning, and the pipeline's design and operational constraints allowed only daily data aggregations and forced a one-day gap between the last training set observation and the target day. Overcoming these constraints would enable more complex approaches, such as expanding hybridizations, applying neural networks, or further data explorations.

7 References

Akiba, Takuya, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. 'Optuna: A Next-Generation Hyperparameter Optimization Framework'. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2623–31. Anchorage AK USA: ACM. https://doi.org/10.1145/3292500.3330701.

Aldor-Noiman, Sivan, Paul D. Feigin, and Avishai Mandelbaum. 2009. 'Workload Forecasting for a Call Center: Methodology and a Case Study'. *The Annals of Applied Statistics* 3 (4). https://doi.org/10.1214/09-AOAS255.

Ardabili, Sina, Amir Mosavi, and Annamária R. Várkonyi-Kóczy. 2020. 'Advances in Machine Learning Modeling Reviewing Hybrid and Ensemble Methods'. In *Engineering for Sustainable Future*, edited by Annamária R. Várkonyi-Kóczy, 101:215–27. Lecture Notes in Networks and Systems. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-36841-8_21.

Armstrong, Jon Scott. 2001. *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Vol. 30. Springer.

Assimakopoulos, V., and K. Nikolopoulos. 2000. 'The Theta Model: A Decomposition Approach to Forecasting'. *International Journal of Forecasting* 16 (4): 521–30. https://doi.org/10.1016/S0169-2070(00)00066-2.

Bates, J. M., and C. W. J. Granger. 1969. 'The Combination of Forecasts'. *Journal of the Operational Research Society* 20 (4): 451–68. https://doi.org/10.1057/jors.1969.103.

Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. 1992. 'A Training Algorithm for Optimal Margin Classifiers'. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–52. Pittsburgh Pennsylvania USA: ACM. https://doi.org/10.1145/130385.130401.

Box, George E. P., Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. 2016. *Time Series Analysis: Forecasting and Control*. Fifth edition. Wiley Series in Probability and Statistics. Hoboken, New Jersey: Wiley.

Brown, RG. 1959. 'Statistical Forecasting for Inventory Control'. *McGraw-Hill Google Schola* 2:443–73.

Chen, Tianqi, and Carlos Guestrin. 2016. 'XGBoost: A Scalable Tree Boosting System'. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–94. San Francisco California USA: ACM. https://doi.org/10.1145/2939672.2939785.

Clemen, Robert T. 1989. 'Combining Forecasts: A Review and Annotated Bibliography'. *International Journal of Forecasting* 5 (4): 559–83. https://doi.org/10.1016/0169-2070(89)90012-5.

Dickey, David A., and Wayne A. Fuller. 1979. 'Distribution of the Estimators for Autoregressive Time Series with a Unit Root'. *Journal of the American Statistical Association* 74 (366a): 427–31. https://doi.org/10.1080/01621459.1979.10482531.

Friedman, Jerome H. 2001. 'Greedy Function Approximation: A Gradient Boosting Machine'. *Annals of Statistics*, 1189–1232.

Gans, Noah, Ger Koole, and Avishai Mandelbaum. 2003. 'Telephone Call Centers: Tutorial, Review, and Research Prospects'. *Manufacturing & Service Operations Management* 5 (2): 79–141. https://doi.org/10.1287/msom.5.2.79.16071.

Hajirahimi, Zahra, and Mehdi Khashei. 2019. 'Hybrid Structures in Time Series Modeling and Forecasting: A Review'. *Engineering Applications of Artificial*

Intelligence 86 (November):83–106. https://doi.org/10.1016/j.engappai.2019.08.018.

———. 2023. 'Hybridization of Hybrid Structures for Time Series Forecasting: A Review'. *Artificial Intelligence Review* 56 (2): 1201–61. https://doi.org/10.1007/s10462-022-10199-0.

Holt, Charles C. 2004. 'Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages'. *International Journal of Forecasting* 20 (1): 5–10. https://doi.org/10.1016/j.ijforecast.2003.09.015.

Ibrahim, Rouba, Han Ye, Pierre L'Ecuyer, and Haipeng Shen. 2016. 'Modeling and Forecasting Call Center Arrivals: A Literature Survey and a Case Study'. *International Journal of Forecasting* 32 (3): 865–74. https://doi.org/10.1016/j.ijforecast.2015.11.012.

Jaganathan, Srihari, and P.K.S. Prakash. 2020. 'A Combination-Based Forecasting Method for the M4-Competition'. *International Journal of Forecasting* 36 (1): 98–104. https://doi.org/10.1016/j.ijforecast.2019.03.030.

Ke, Guolin, Qi Meng, Thomas Finely, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. 'LightGBM: A Highly Efficient Gradient Boosting Decision Tree'. In *Advances in Neural Information Processing Systems 30 (NIP 2017)*. https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/.

Kwiatkowski, Denis, Peter C.B. Phillips, Peter Schmidt, and Yongcheol Shin. 1992. 'Testing the Null Hypothesis of Stationarity against the Alternative of a Unit Root'. *Journal of Econometrics* 54 (1–3): 159–78. https://doi.org/10.1016/0304-4076(92)90104-Y.

Makridakis, Spyros, and Michèle Hibon. 2000. 'The M3-Competition: Results, Conclusions and Implications'. *International Journal of Forecasting* 16 (4): 451–76. https://doi.org/10.1016/S0169-2070(00)00057-1.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2020. 'The M4 Competition: 100,000 Time Series and 61 Forecasting Methods'. *International Journal of Forecasting* 36 (1): 54–74. https://doi.org/10.1016/j.ijforecast.2019.04.014.

———. 2022. 'M5 Accuracy Competition: Results, Findings, and Conclusions'. *International Journal of Forecasting* 38 (4): 1346–64. https://doi.org/10.1016/j.ijforecast.2021.11.013.

Makridakis, Spyros, and Robert L. Winkler. 1983. 'Averages of Forecasts: Some Empirical Results'. *Management Science* 29 (9): 987–96. https://doi.org/10.1287/mnsc.29.9.987.

Nie, Hongzhan, Guohui Liu, Xiaoman Liu, and Yong Wang. 2012. 'Hybrid of ARIMA and SVMs for Short-Term Load Forecasting'. *Energy Procedia* 16:1455–60. https://doi.org/10.1016/j.egypro.2012.01.229.

Petropoulos, Fotios, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K. Barrow, Souhaib Ben Taieb, Christoph Bergmeir, et al. 2022. 'Forecasting: Theory and Practice'. *International Journal of Forecasting* 38 (3): 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001.

Sapankevych, Nicholas, and Ravi Sankar. 2009. 'Time Series Prediction Using Support Vector Machines: A Survey'. *IEEE Computational Intelligence Magazine* 4 (2): 24–38. https://doi.org/10.1109/MCI.2009.932254.

Savitzky, Abraham., and M. J. E. Golay. 1964. 'Smoothing and Differentiation of Data by Simplified Least Squares Procedures.' *Analytical Chemistry* 36 (8): 1627–39. https://doi.org/10.1021/ac60214a047.

Sina, Lennart B., Cristian A. Secco, Midhad Blazevic, and Kawa Nazemi. 2023. 'Hybrid Forecasting Methods—A Systematic Review'. *Electronics* 12 (9): 2019. https://doi.org/10.3390/electronics12092019.

Taylor, James W. 2008. 'A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center'. *Management Science* 54 (2): 253–65. https://doi.org/10.1287/mnsc.1070.0786.

Winters, Peter R. 1960. 'Forecasting Sales by Exponentially Weighted Moving Averages'. *Management Science* 6 (3): 324–42. https://doi.org/10.1287/mnsc.6.3.324.

Yule, Geroge Udny. 1927. 'VII. On a Method of Investigating Periodicities Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers'. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226 (636–646): 267–98. https://doi.org/10.1098/rsta.1927.0007.

8 Appendix

Figure 3: Simplification of the modeling pipeline for one day

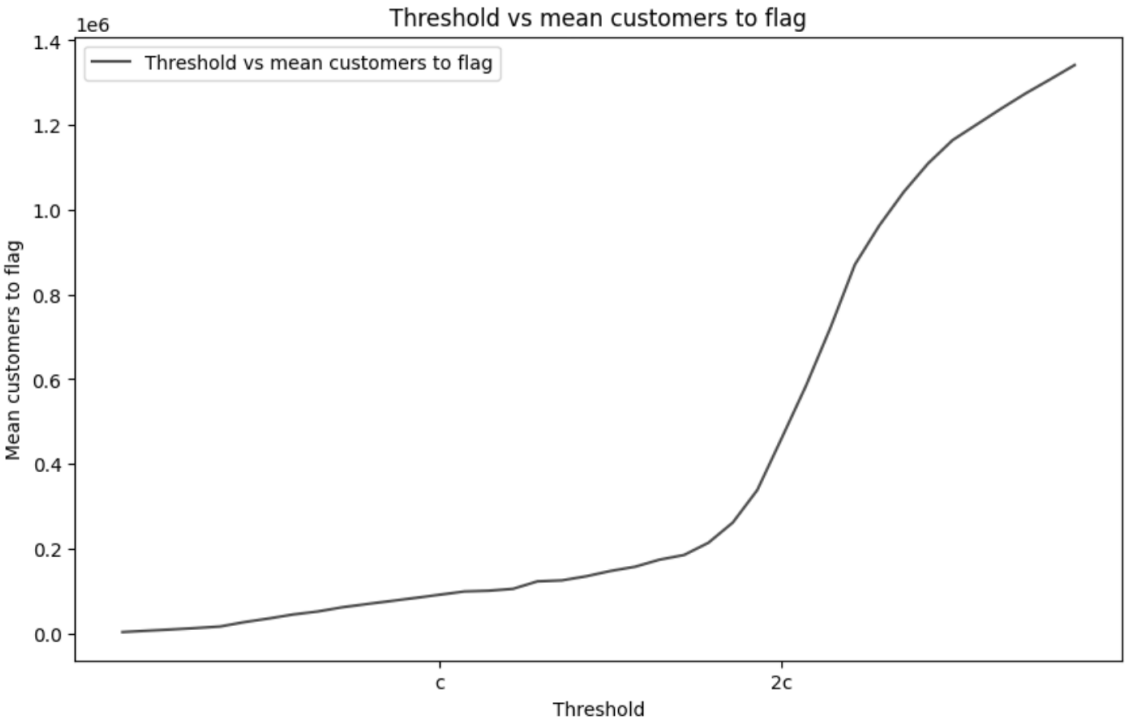

Figure 4: Relationship of customers to flag to reach c to incoming calls to the technical line

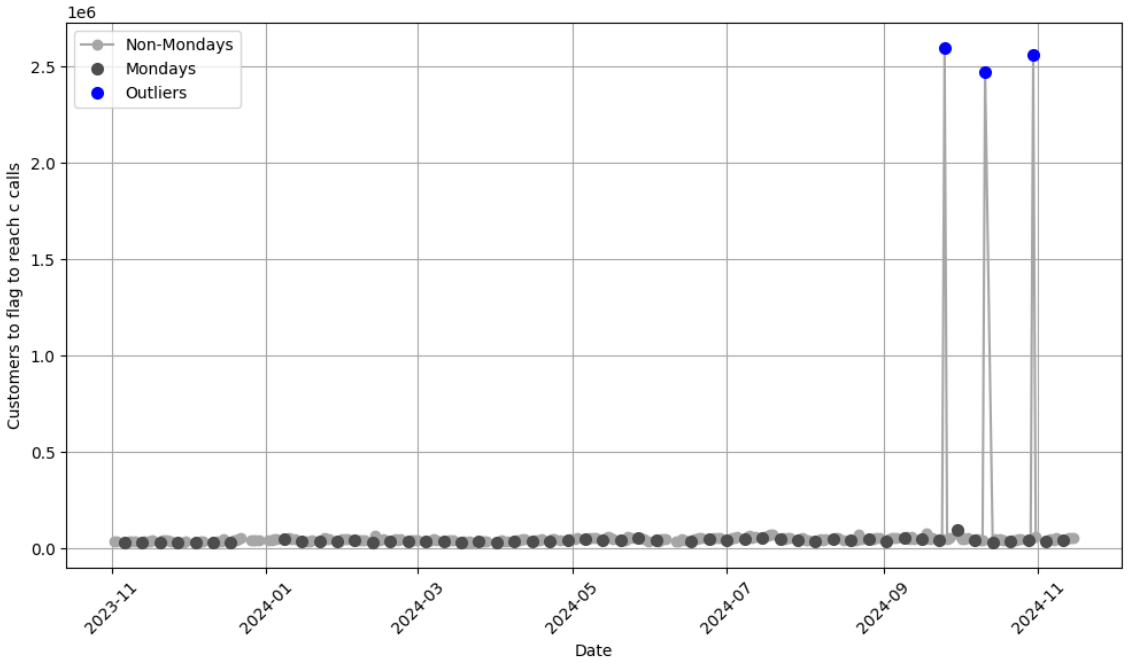

Figure 5: All observations showing the three outliers due to operational reasons

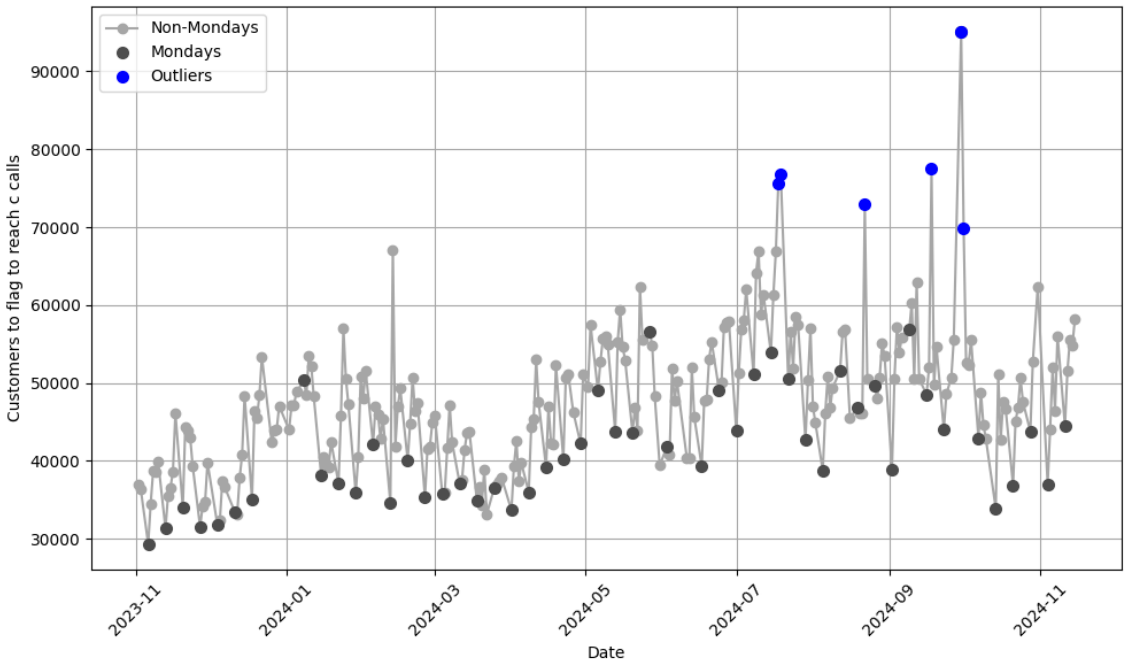

Figure 6: All observations

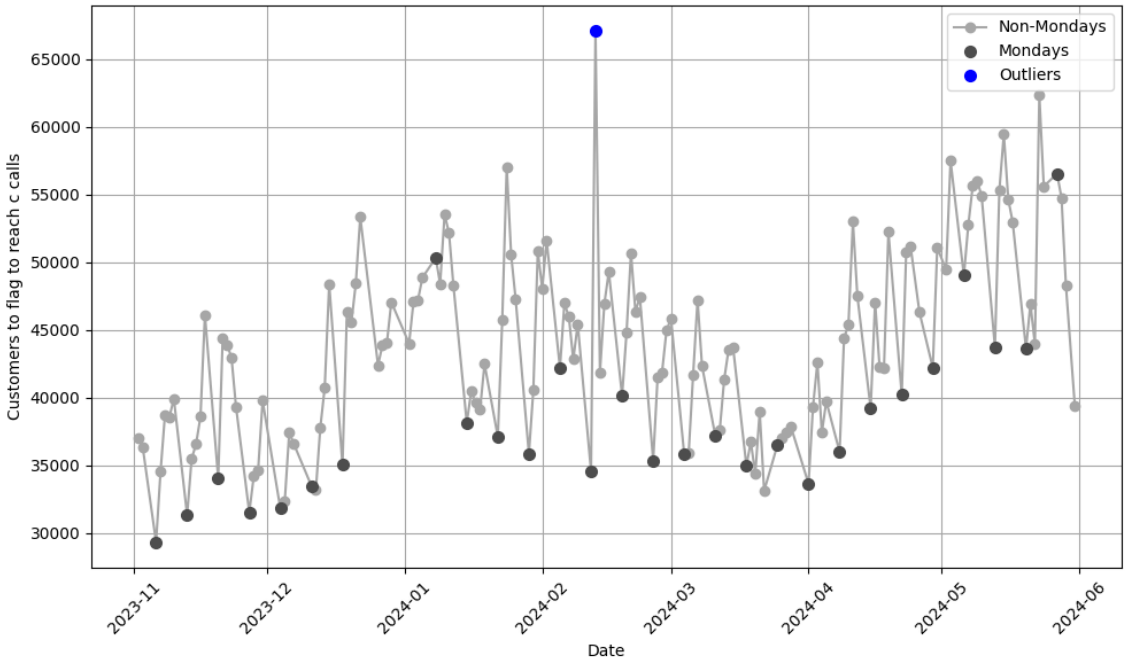

Figure 7: Training set

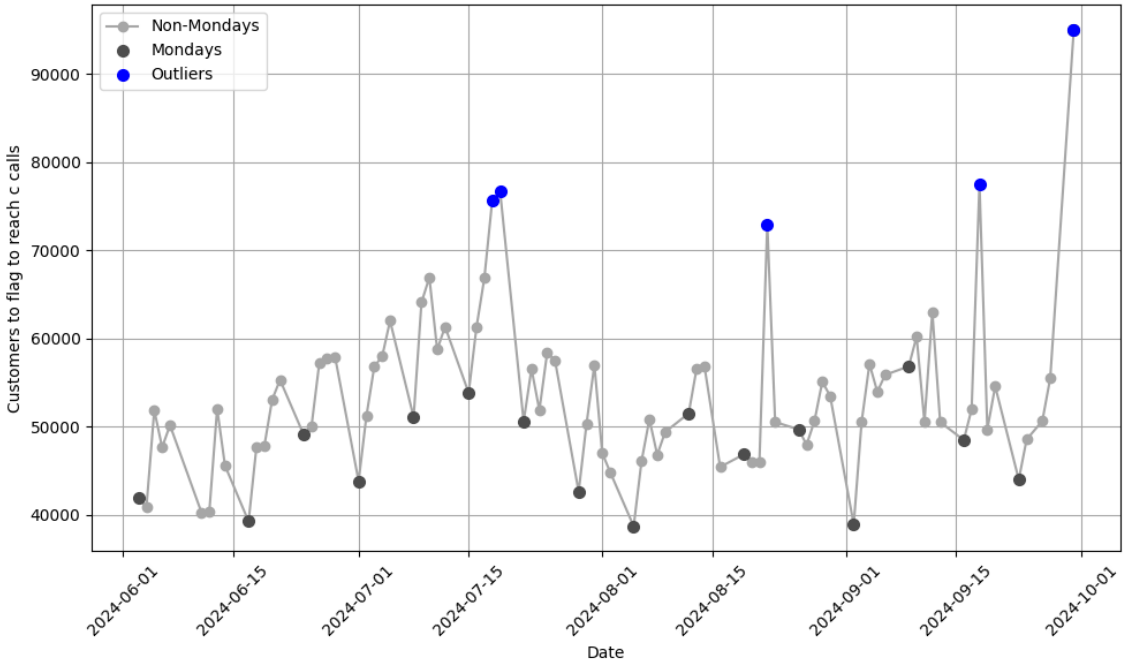

Figure 8: Validation Set

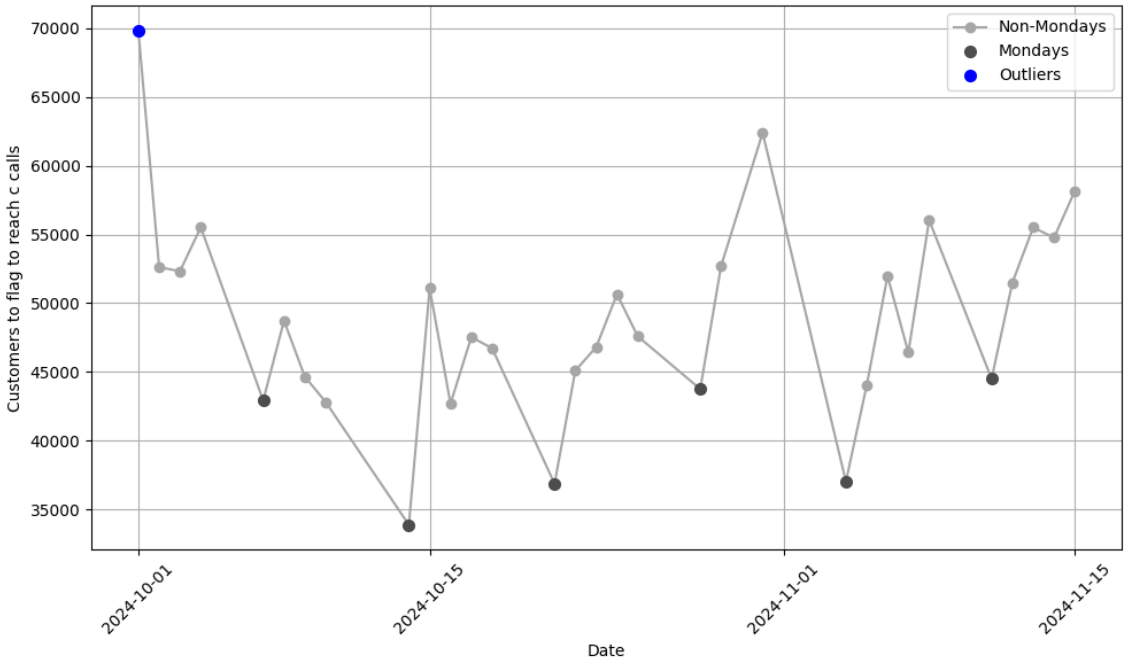

Figure 9: Holdout set

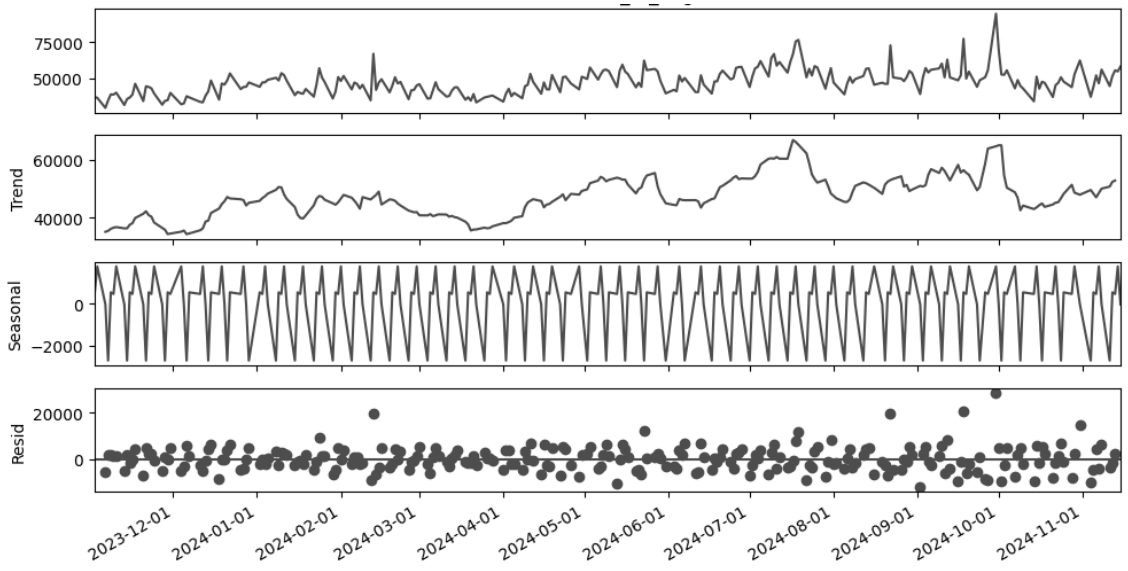

Figure 10: Weekly decomposition of the data

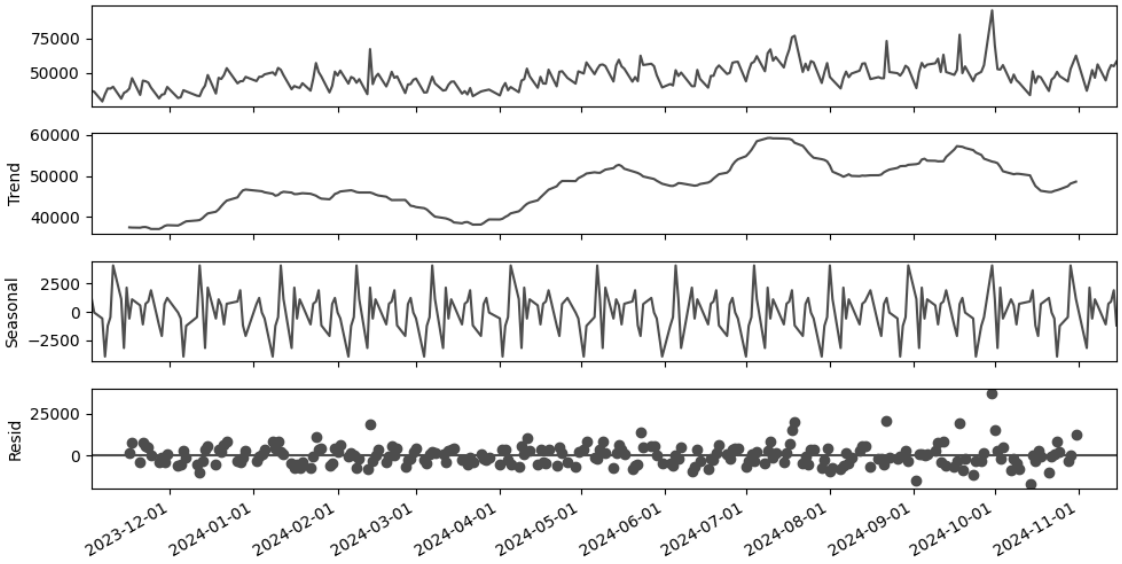

Figure 11: Monthly decomposition of the data

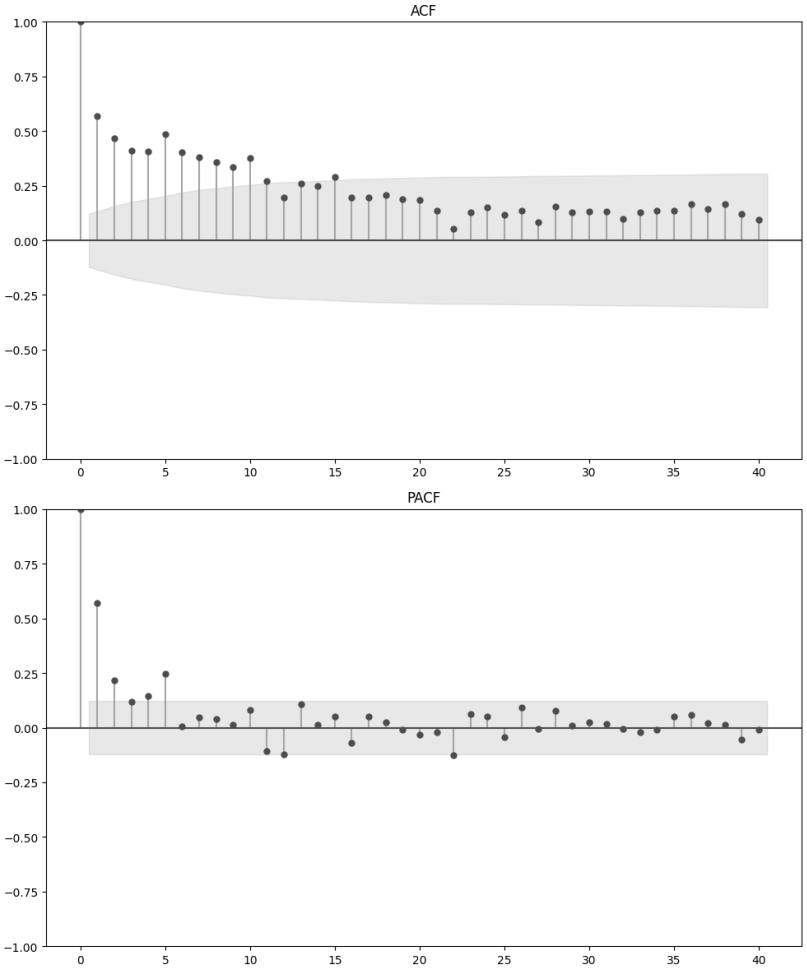

Figure 12: ACF & PACF plots

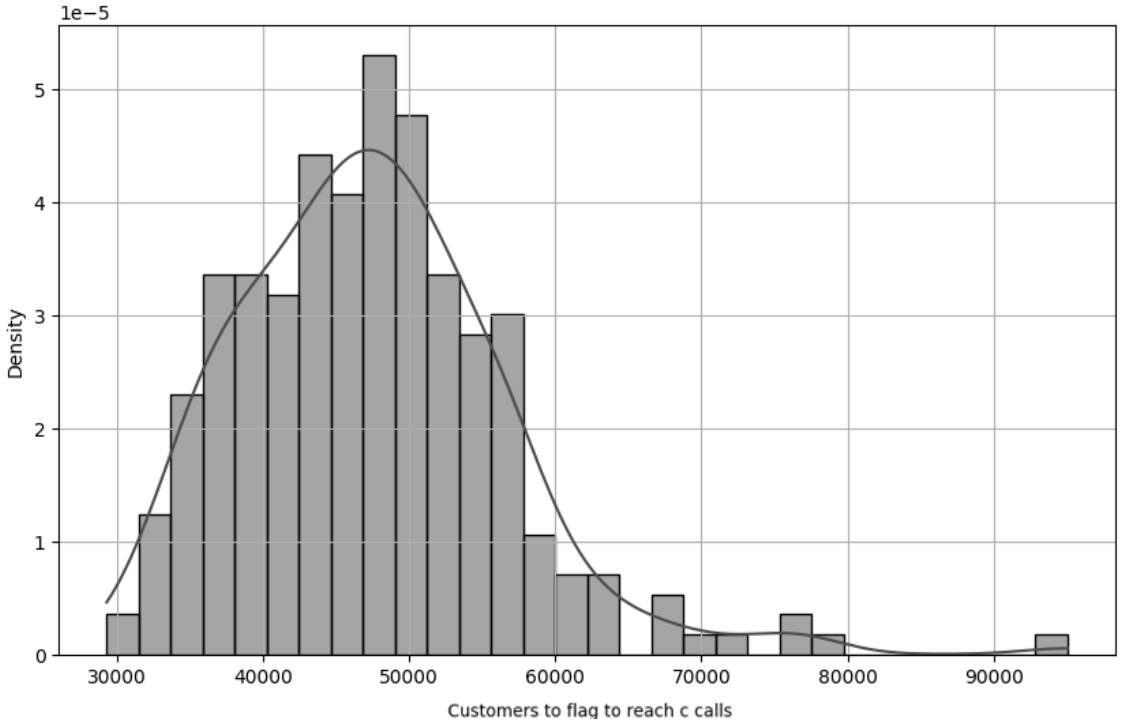

Figure 13: Histogram with density curve of the data

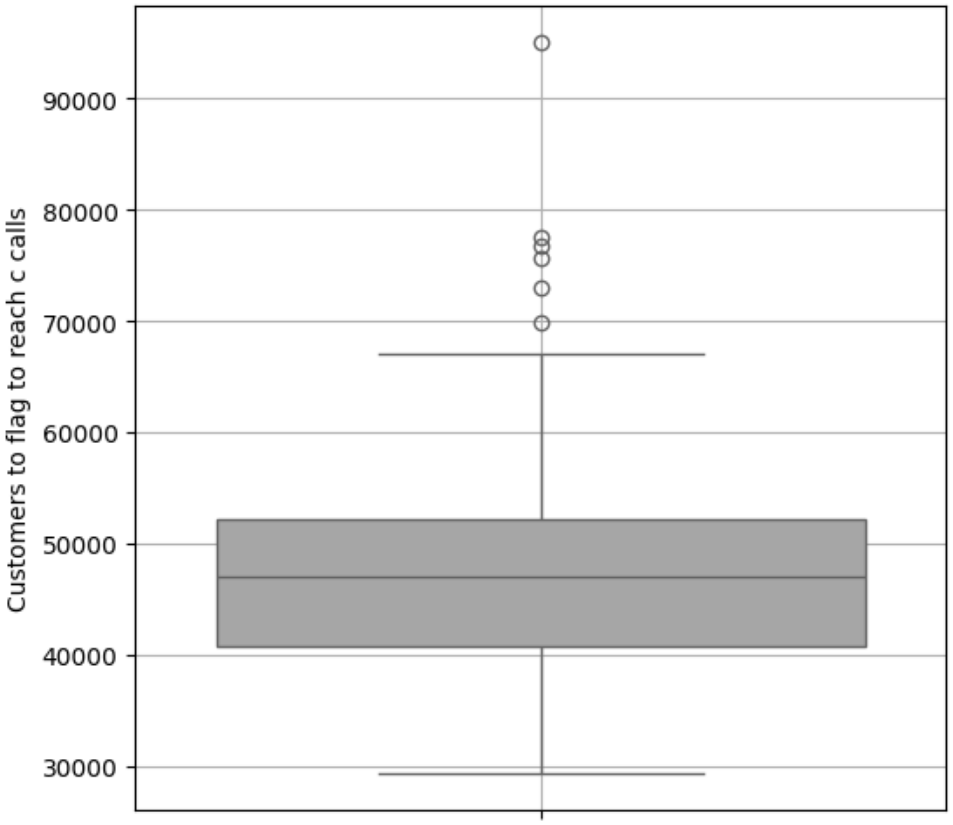

Figure 14: Boxplot of the data

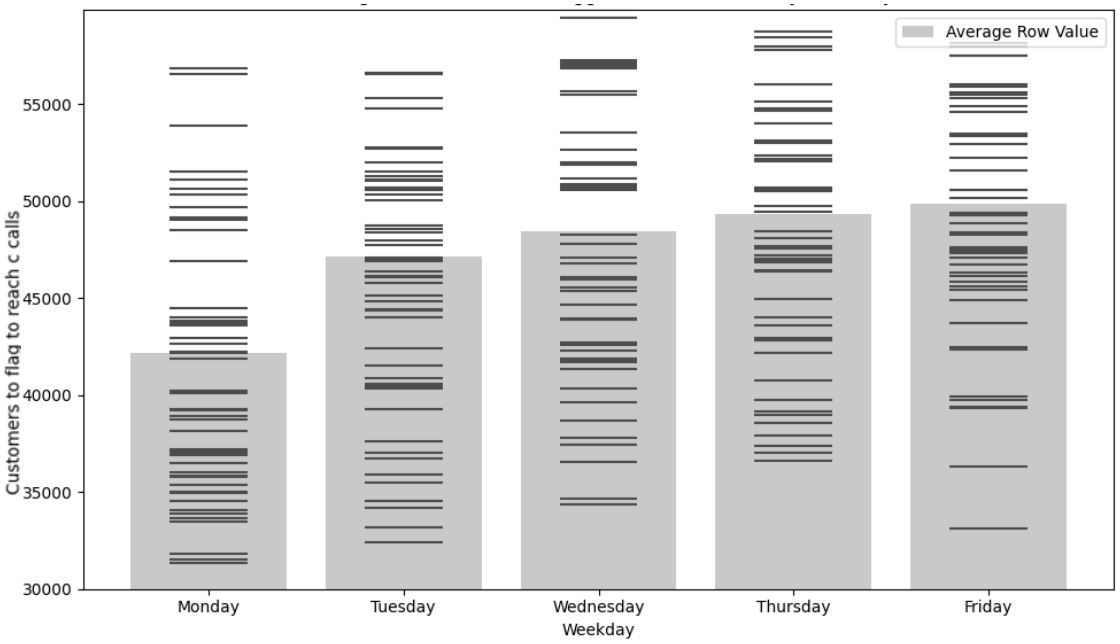

Figure 15: Averages per weekday of customers to flag to reach c, displaying weekly seasonality

Table 3: Ranges and optimal hyperparameters of all applied models

Models	**Error**	**Hyperparameter ranges**	**Best hyperparameters**
All Models		Train set size: 5 to 140 steps of 1 IQR multiplier outlier removal: 1 to 10 steps of 0.1	
Average	MAPE	As seen above	Train set size: 50 IQR multiplier outlier removal: 1.6
Average	MSE	As seen above	Train set size: 5 IQR multiplier outlier removal: 1.5
Weekday average	MAPE	As seen above	Train set size: 54 IQR multiplier outlier removal: 1.6
Weekday average	MSE	As seen above	Train set size: 52 IQR multiplier outlier removal: 1.1
Smoothened Weekday Average	MAPE	Window length of Savitzky-Golay filter: 1 to 90 Polyorder of Savitzky-Golay Filter: 1 to 15	Train set size: 54 IQR multiplier outlier removal: 1.3 Window length of Savitzky-Golay filter : 11 Polyorder if Savitzky-Golay Filter : 8
Smoothened Weekday Average	MSE	As seen above	Train set size: 55 IQR multiplier outlier removal: 1.3 Window length of Savitzky-Golay filter : 7 Polyorder if Savitzky-Golay Filter : 6
SARIMA	MAPE	p, d, & q: 1 to 3 steps of 1 P, D, Q: 1 to 3 steps of 1 s (period) = 5 or 20	Train set size: 69 IQR multiplier outlier removal: 1.9 p: 1, d: 0, q: 2, P: 1, D: 0, Q: 1, s: 5
SARIMA	MSE	As seen above	Train set size: 60 IQR multiplier outlier removal: 1.9 p: 1, d: 0, q: 1, P: 1, D: 0, Q: 2, s: 5

Theta	MAPE	deseasonalize: Boolean use_test: Boolean method: auto, additive or multiplicative difference: Boolean period: 5 or 20	Train set size: 104 IQR multiplier outlier removal: 1.4 deseasonalize: True use_test: 0 method: auto difference: False period: 5
Theta	MSE	As seen above	Train set size: 134 IQR multiplier outlier removal: 1.8 deseasonalize: True use_test: True method: auto difference: True period: 5
ES	MAPE	alpha: 0.01 to 1 continuous beta: 0 to 1 continuous gamma: 0 to 1 continuous seasonal: add, mul or None trend: add, mul, None seasonal periods: 5 or 20	Train set size: 83 IQR multiplier outlier removal: 1.1 alpha: 0.46707694093801616 beta: 0 gamma: 0.004822396362337833 seasonal: mul trend: None seasonal periods: 5
ES	MSE	As seen above	Train set size: 70 IQR multiplier outlier removal: 1.9 alpha: 0.16384305990684625 beta: 0.6596058471015328 gamma: 0.3290353416446719 seasonal: add trend: None seasonal periods: 5
LightGBM	MAPE	boosting_type: gbdt or rf objective: gamma, regression, regression_l1 or huber, fair metric: MSE or MAPE max_depth: 3 to 20 in steps of 1 learning_rate: 0.1 to 0.3 continuous n_estimators: 50 to 1000 in steps of 1 num_leaves: 31 to 511 in steps of 1 min_child_weight': 0.0001 to 1 continuous feature_fraction: 0.6 to 1 continuous bagging_fraction: 0.6, 1 continuous bagging_freq: 0 to 5 in steps of 1	Train set size: 128 IQR multiplier outlier removal: 8.5 boosting_type: gbdt objective: regression_l1 metric: MAPE max_depth: 10 learning_rate: 0.212290115674774 n_estimators: 933 num_leaves: 299 min_child_weight: 0.439665848228 feature_fraction 0.65576006249106 bagging_fraction: 0.9040606219977 bagging_freq: 3 lambda_l1: 4.383297281143508 lambda_l2: 3.7355182951836086 min_split_gain: 0 max_bin: 424

		lambda_l1: 0 to 5.1 continuous lambda_l2: 0, 5.1 continuous min_split_gain: 0, 0.2 continuous max_bin: 255 to 1000 in steps of 1	
LightGBM	MSE	As seen above	Train set size: 131 IQR multiplier outlier removal: 8.1 boosting_type: gbdt objective: regression_l1 metric: MSE max_depth: 15 learning_rate: 0.2956495452328944 n_estimators: 358 num_leaves: 162 min_child_weight': 0.546341438699 feature_fraction: 0.68016073442241 bagging_fraction: 0.9561860996462 bagging_freq: 2 lambda_l1: 1.5398950692154563 lambda_l2: 0.017719961213482627 min_split_gain: 0 max_bin: 331
LightGBM recursive	MAPE	As seen above	Train set size: 134 IQR_multiplier_train = 3.7 boosting_type: gbdt objective: gamma metric: MAPE max_depth: 6 learning_rate: 0.2308704733396836 n_estimators: 636 num_leaves: 89 min_child_weight: 0.868715703852 feature_fraction: 0.71302501051863 bagging_fraction: 0.7471335675595 bagging_freq: 5 lambda_l1: 0.9122993501626743 lambda_l2: 0.9806665339853291 min_split_gain: 0 max_bin: 704
LightGBM recursive	MSE	As seen above	Train set size: 133 IQR multiplier train: 3.6 boosting_type: gbdt objective: regression_l1 metric: MSE max_depth: 15 learning_rate: 0.29564954523289444 n_estimators: 358 num_leaves: 162

			min_child_weight: 0.5463414386984622 feature_fraction: 0.6801607344224033 bagging_fraction: 0.9561860996462354 bagging_freq: 2 lambda l1: 1.5398950692154563 lambda_l2: 0.017719961213482627 min_split_gain: 0 max_bin: 331 tweedie_variance_power: 1.0285251982425723
SVM	MAPE	kernel: poly, rbf or sigmoid C: 1e-4 to 1e4 continuous epsilon: 0.01, 1.0 Continuous float degree: 2 to 5 in steps of one, only for poly kernel gamma: 1e-5 to 1e2 continuous coef0: -10 to 10 continuous float only for poly and sigmoid kernel	Train set size: 124 IQR multiplier train: 1.2 kernel: rbf C: 9477.4022822747 epsilon: 0.5579095982652081 degree: 5 gamma: 0.007653474502926799 coef0: -3.0560534751372765
SVM	MSE	As seen above	Train set size: 119 IQR multiplier train: 1.6 kernel: poly C: 6997.648460789635 epsilon: 0.45154502556905796 degree: 3 gamma: 0.0007200816767883813 coef0: -6.519615514829753
SVM recursive	MAPE	As seen above	Train set size: 139 IQR multiplier train: 1.3 kernel: poly C: 5763.137054754013 epsilon: 0.4237460123541047 degree: 5 gamma: 0.0001811438255404319 5 coef0: 2.3048667763008144
SVM recursive	MSE	As seen above	Train set size: 129 IQR multiplier train: 1.8 kernel: poly C: 8304.221974143353 epsilon: 0.3896657272715893 degree: 5 gamma: 0.00006409279661943258 coef0: 7.484084736141426

Table 4: Performance evaluation of all applied models ordered by MAPE ascending

Model	MAPE	MSE	Mae	Mapc
Parallel Pareto VarCov	10.52	41802239	5008	4.69
Theta MAPE	10.59	47285069	5071	5.92
Parallel below 13 VarCov	10.79	44183548	5096	3.70
Weekdays average MAPE	10.93	41112555	5139	6.44
Parallel Pareto linear VarCov	10.96	43623707	5143	3.34
Weekday average smooth MSE	11.20	43278815	5254	5.56
Weekday average smooth MSE	11.26	44444199	5268	5.79
Parallel all VarCov	11.29	51035801	5352	4.15
Parallel all simple average	11.38	51019091	5384	4.10
Series Pareto linear - Pareto non-linear	11.39	40733085	5238	3.39
Weekday average MSE	11.40	43552952	5319	4.76
Theta MSE	11.47	52678275	5501	6.66
Parallel all bayesian	11.49	51457987	5430	3.82
SVM recursive MAPE	11.72	54130469	5446	5.27
Series Parallel non-linear Pareto VarCov	11.85	54199060	5607	4.36
SVM recursive MSE	11.91	55835742	5561	7.13
ES MAPE	11.93	54970382	5717	5.23
SVM MAPE	12.02	55533776	5829	4.53
Weekday average smooth MAPE	12.15	49078471	5674	4.36
Parallel SVM VarCov	12.15	55780723	5763	5.37
Average MAPE	12.62	55412704	5865	0.37
Series Pareto non-linear - Pareto linear	12.80	57416042	6005	5.14
ES MSE	13.21	63803680	6271	7.86
Parallel all NNLS	13.23	64421965	6243	5.20
LightGBM MSE	13.39	68950629	6387	5.64
LightGBM recursive MSE	13.49	70890795	6282	6.87
LightGBM recursive MAPE	13.49	74910450	6268	6.05
SVM MSE	13.99	69846518	6717	6.83
LightGBM MAPE	14.09	73536593	6724	4.78
Average MSE	14.81	81208427	7101	4.46
Series theta - SVM recursive	15.32	89267051	7303	8.34
SARIMA MAPE	15.40	81367878	7273	3.58
SARIMA NOS	15.42	85489146	7359	4.46
SARIMA MSE	16.21	95631142	7722	5.83
Naive	18.33	143282066	8850	18.01
Parallel all OLS	19.09	114579218	8843	9.32

Table 5: Compositions of applied hybrid parallel models

Model	Composition
Parallel Pareto linear	Average MAPE Weekday average MSE Smoothened weekday average MAPE Smoothened weekday average MSE Theta MAPE
Parallel Pareto non-linear	SVM MAPE SVM recursive MAPE
Parallel below 13 MAPE	Average MAPE Weekday average MAPE Weekday average MSE Weekday average smoothened MAPE Weekday average smoothened MSE Theta MAPE Theta MSE ES MAPE SVM MAPE SVM recursive MAPE SVM recursive MSE
Parallel Pareto	Average MAPE Weekday average MSE Smoothened weekday average MAPE Smoothened weekday average MSE SVM MAPE Theta MAPE

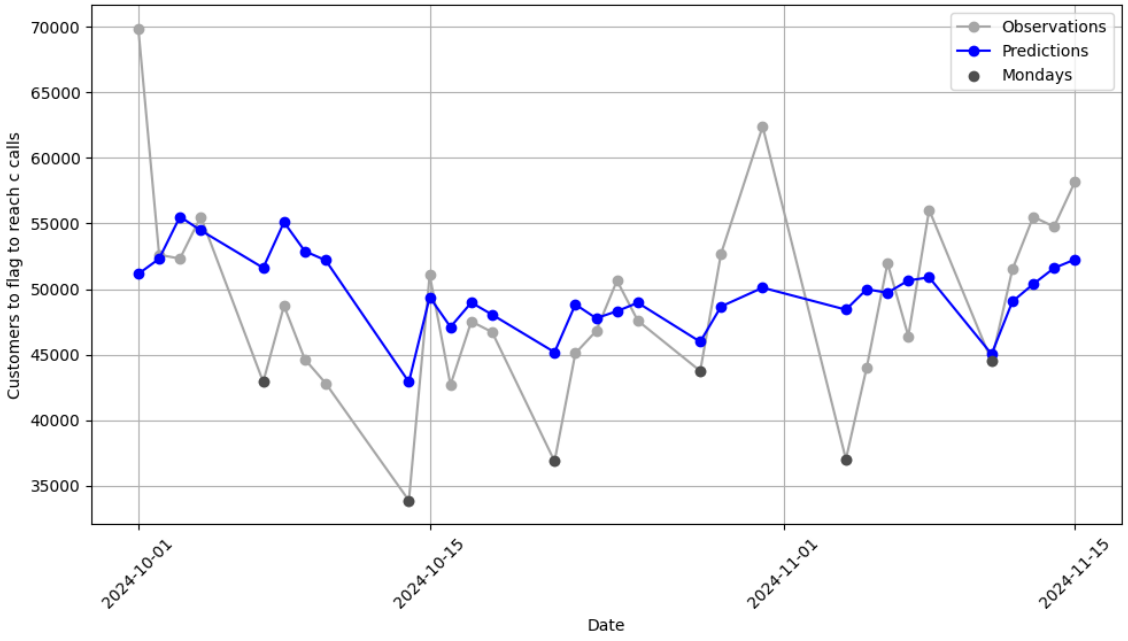

Figure 10: Predictions of the parallel hybrid of the Pareto-efficient models using VarCov for the EW

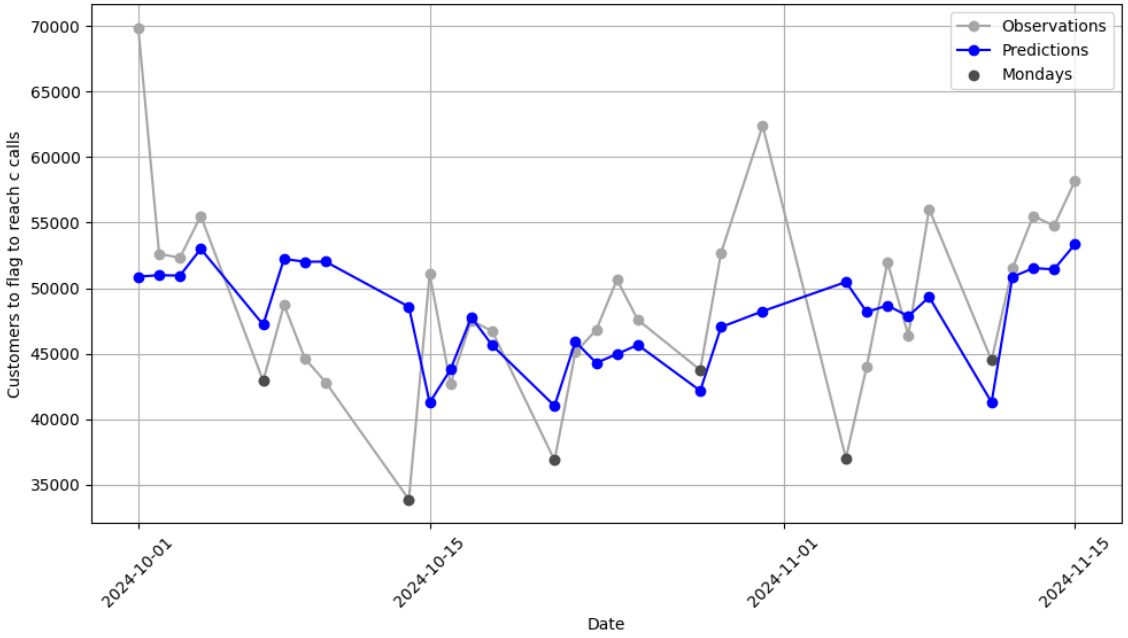

Figure 11: Predictions of the for MAPE optimized theta model for the EW

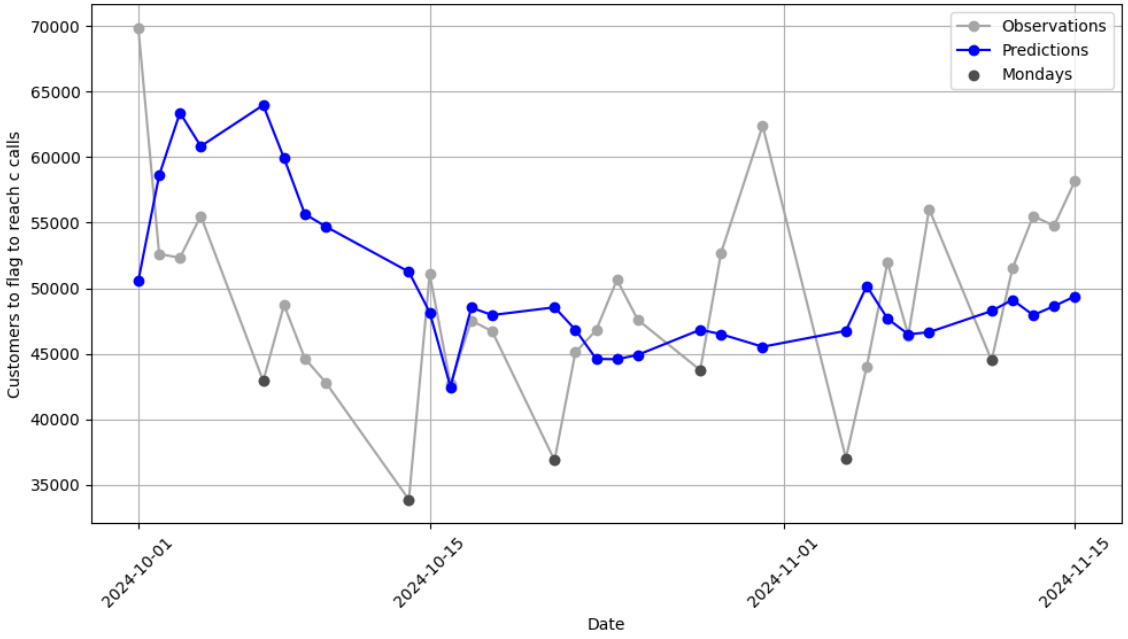

Figure 12: Predictions of the at NOS deployed SARIMA for the EW

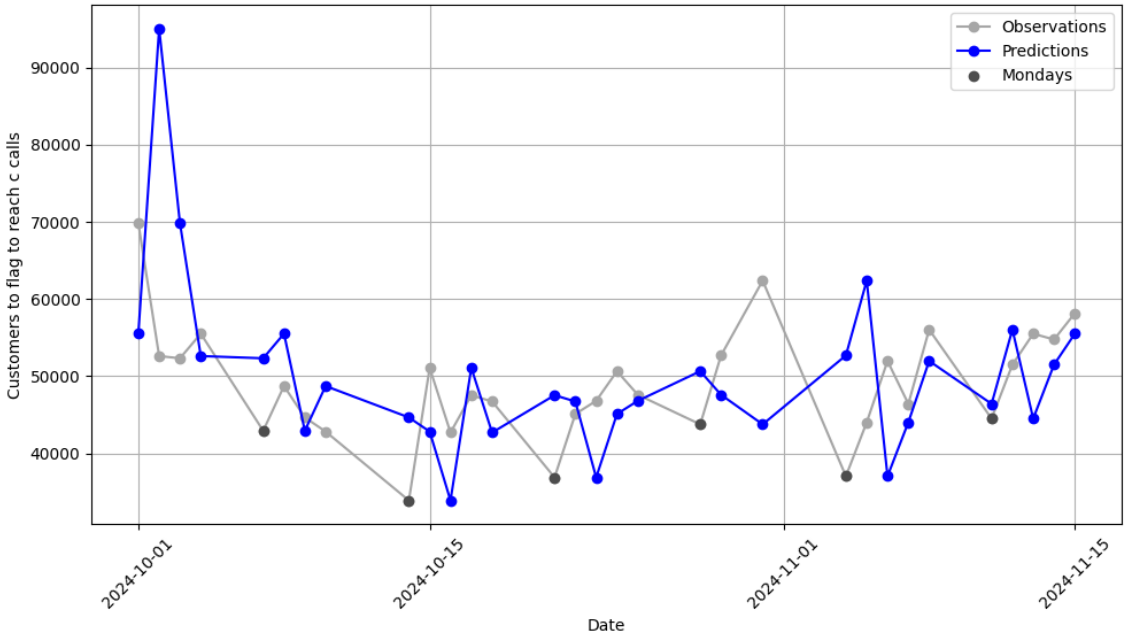

Figure 13: Predictions of the Naive model for the EW

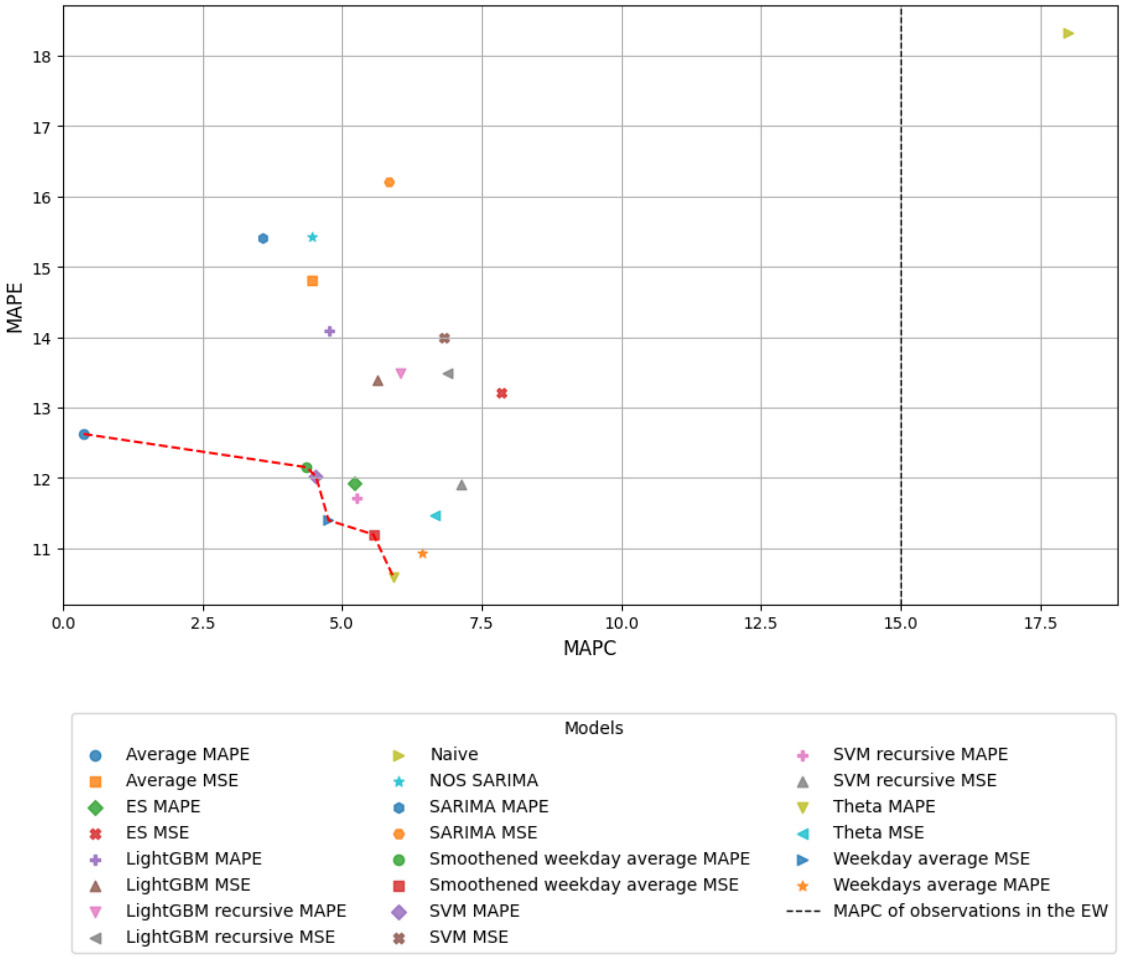

Figure 14: Evaluation and Pareto frontier of all individual models

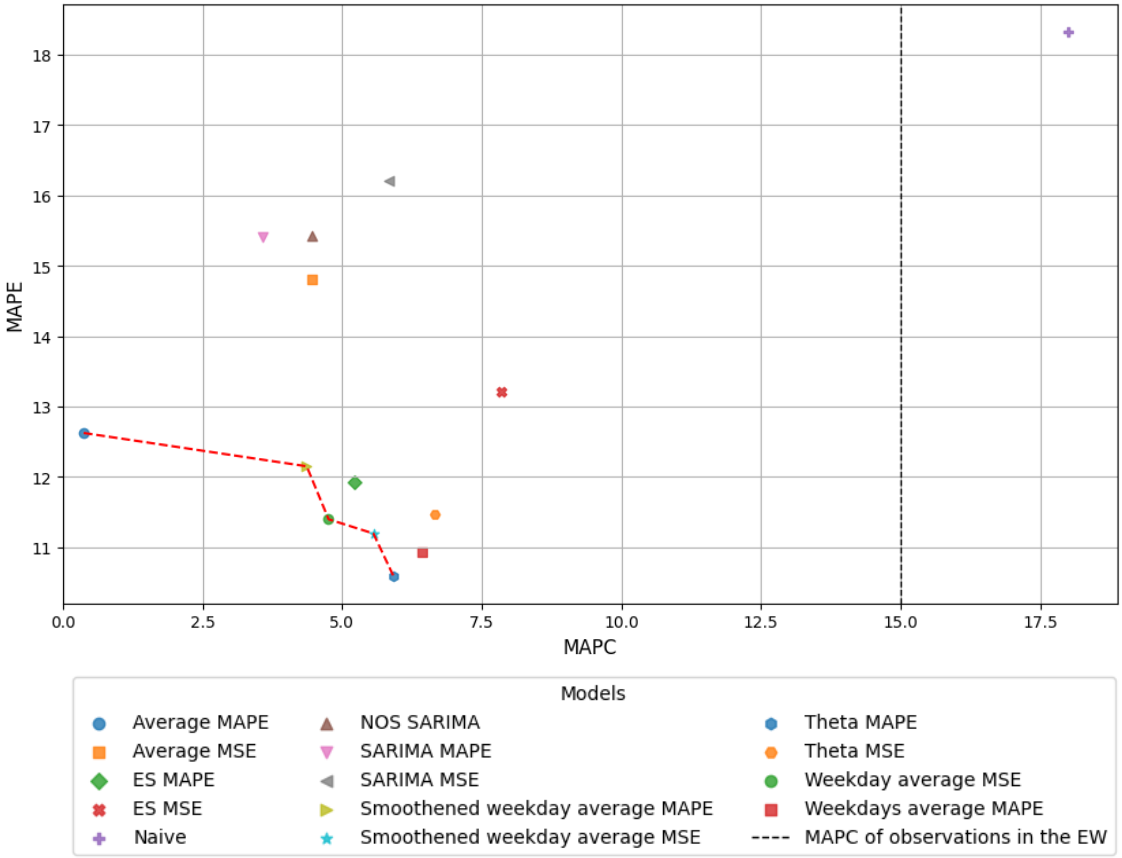

Figure 15: Evaluation and Pareto frontier of linear individual models

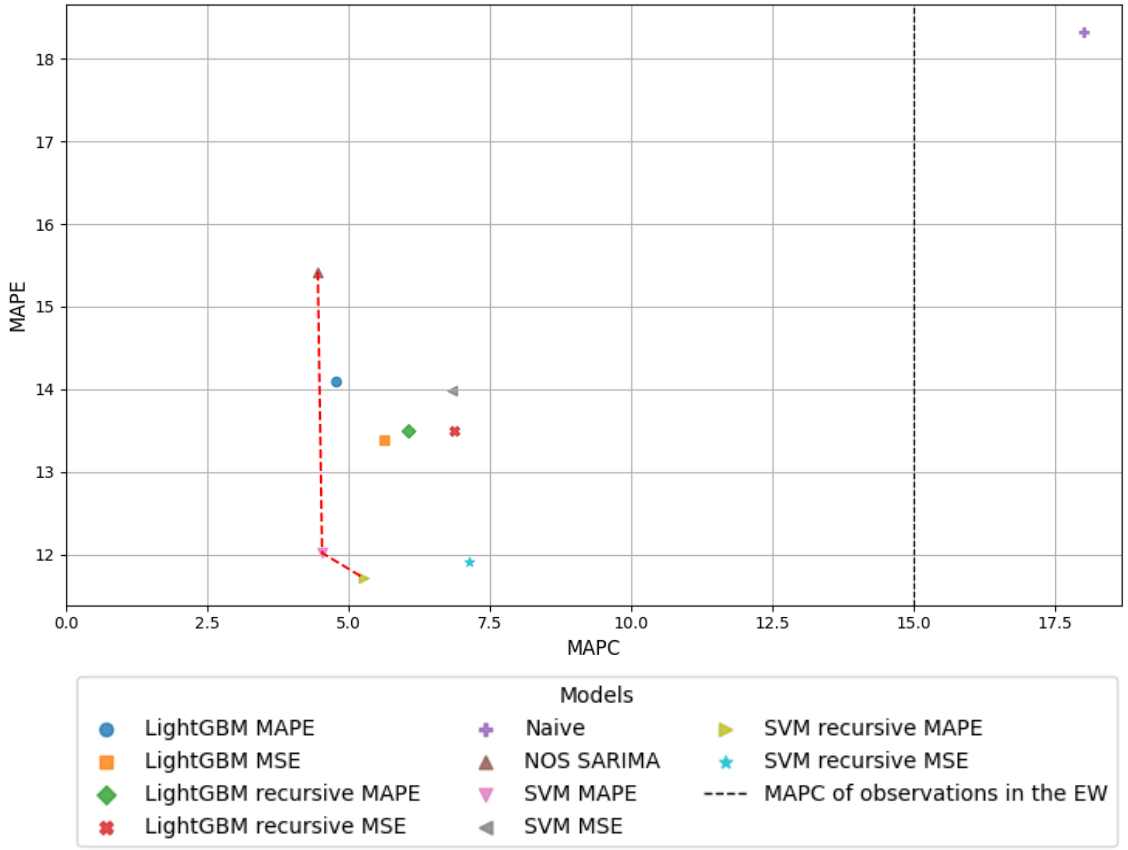

Figure 16: Evaluation and Pareto frontier of non-linear individual models