

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

Harnessing the Wisdom of the Crowd: Ensemble Methods for Time Series Forecasting of Call Center Arrivals

AUTHOR

Hendrik Künnemann

Work project carried out under the supervision of:

Susana Lavado &

Gustavo Pereira

29-01-2025

Abstract

This study extends a time series forecasting project (PBL) on a small dataset by examining ensemble learning, including homogeneous (bagging) and heterogeneous (Dynamic Integration) approaches. While bagging slightly reduces accuracy (MAPE), it improves stability. By incorporating a novel error-based dynamic pairwise correlation strategy to enhance diversity between base-learners, the Dynamic Weighting with Selection method within Dynamic Integration significantly outperforms the baseline, reducing the error metric MAPE by nearly 10% and the stability metric by over 20%. These findings highlight the effectiveness of ensemble learning, particularly DWS, for accurate and reliable forecasting in small datasets.

Keywords: Time Series Forecasting, Ensemble Methods, Bagging, Moving Block Bootstrap, Dynamic Integration, Diversity among Base-learners, Small Dataset

Acknowledgements

I would like to express my heartfelt gratitude to my advisors, Susana Lavado and Gustavo Pereira, for their invaluable guidance and constructive feedback, which were instrumental throughout both the thesis and PBL phases. I am also deeply grateful to my PBL team, João Miranda, João Penedo, Victoria Muguerza, and Tom Göring, for their dedication during the PBL project and for their contributions in turning our work into a joint thesis. Lastly, I sincerely thank my family, girlfriend, and friends for their unwavering support and encouragement throughout this journey. Each of you played a crucial role in this achievement, and I truly appreciate it.

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).

Contents

1	Group Part	4
1.1	Introduction	4
1.2	Context	5
1.3	Dataset	7
1.4	Literature Review	8
1.4.1	Time Series Forecasting	8
1.4.2	Time Series Forecasting at Call Centers	12
1.4.3	Time Series Forecasting in Limited Size Datasets	12
1.5	Project-Based Learning Project	13
1.5.1	Target Variable and Data Preprocessing	13
1.5.2	Feature Engineering	14
1.5.3	Methodology	15
1.5.3.1	Target Capping	15
1.5.3.2	Introduction of the Models Deployed	16
1.5.3.3	Model Evaluation Criteria	20
1.5.4	Results from PBL	26
1.5.4.1	Deployable Solution	28
1.5.4.2	Build-Onto Solution	29
2	Individual Part	30
2.1	Literature Review	31
2.1.1	Introduction to Ensemble Methods	31
2.1.2	Why do ensemble methods work	33
2.1.3	Main characteristics of a good ensemble	34
2.1.3.1	How to ensure diversity	34
2.1.4	Aggregation Methods	35
2.1.5	Selection of Key Ensemble Methods for This Study	36
2.1.5.1	Bagging for Time Series Forecasting	36
2.1.5.2	Dynamic Integration for Time Series Forecasting	37

2.2	Methodology	38
2.2.1	Base-learner Choice	38
2.2.2	Bagging	39
2.2.3	Dynamic Integration	40
2.2.4	Hyperparameter tuning	41
2.3	Results & Discussion	41
2.3.1	Bagging Results	42
2.3.2	Dynamic Integration Results	43
2.3.3	Final Model Decision and Performance Evaluation	43
2.4	Limitations & Future Work	46
2.5	Conclusion	47
A	Appendix	59
A.1	Visualization of the Representation Problem	59
A.2	Visualization of the MBB approach	60
A.3	Choice for Base-Learners	61
A.3.1	Base-Learners Bagging	61
A.3.2	Base-Learners Dynamic Integration	61
A.4	Visualization of the rolling window approach	63
A.5	Gridsearch Parameters	65
A.6	Results	66
A.6.1	Bagging Results	66
A.6.2	Dynamic Integration Results	67
A.6.3	Final Model Decision and Performance Evaluation	68

1 Group Part

1.1 Introduction

In today's competitive telecommunications industry, customer satisfaction is essential for retaining a loyal client base. To maintain strong connections with customers, organizations increasingly rely on call centers (Purba and Hidayati 2020). A well-managed call center can greatly enhance an organization's overall performance by optimizing key metrics such as "service level, cost per contact, customer satisfaction, average handle time, first call resolution, abandon rate, average waiting time, and occupancy rate" (Płaza and Pawlik 2021).

NOS, a leading telecommunications operator in Portugal, was established in 2014 through the merger of ZON and Optimus. Recognizing the strategic importance of call center operations, NOS has made significant investments in data science and AI, among them projects to optimize their call center operations. One such initiative was the Project-Based Learning (PBL) in collaboration with NOVA SBE, where a predictive modeling framework was developed to improve key call center metrics. This project provided an opportunity to explore forecasting techniques and operational improvements, focusing on balancing predictive accuracy and stability. The PBL project served as the foundation for this thesis, providing a starting point for predictive modeling and insights into effective call center management strategies.

Building upon the findings from the PBL project, this thesis expands the modeling component by introducing ensemble methods and evaluating various techniques to improve predictive accuracy. It begins by introducing the problem at hand and providing a description of the dataset. Following this, a literature review on time series forecasting is presented, outlining existing methods and their strengths. This leads into the methodology and results from the Project-Based Learning component, which serve as the foundation for the subsequent individual research.

The individual part starts with an introduction to ensemble methods, explaining their fundamental principles, why they are effective, and the key characteristics of a well-constructed ensemble. This is followed by a methodology section detailing the implementation of various ensemble

techniques, including the selection of base learners, bagging, dynamic integration, and hyperparameter tuning for performance optimization. The results are then presented and analyzed, comparing the performance of ensemble methods against the Project-Based Learning results. Finally, the thesis concludes with a discussion of the study's limitations and potential directions for future research, highlighting opportunities to further improve predictive accuracy in time series forecasting.

1.2 Context

NOS' call center is divided into two distinct lines: the general and the technical line. The general line handles a wide range of customer inquiries, including billing questions, contract changes and other non-technical issues. The technical line, on the other hand, specializes in resolving technical issues, such as network issues, service interruptions or other technical difficulties. One of the key challenges within their technical line lies in managing the volume of clients who repeatedly contact the call center to address ongoing issues. These recurrent callers, here referred to as "relapsing callers", not only burden the call center's operations but also pose a significant risk of client dissatisfaction and potential churn. To address this pressing issue, NOS has established a specialized unit, the Detractor Squad, dedicated to handling calls from clients with a history of unresolved issues on the technical line. This unit is specifically designed to manage both relapsing clients and first-time callers who present particularly complex problems. By providing tailored solutions, the Detractor Squad aims to resolve these challenging cases and prevent further relapsing.

Managing the workload of the Detractor Squad effectively is critical. By resolving customer issues and preventing future calls customer satisfaction and retention are enhanced, supporting NOS' broader goal of maintaining a competitive edge in the telecommunications sector. Beyond that, there are also financial benefits, as the Detractor Squad contributes to significant cost savings.

A significant challenge in this process is that relapsing clients do not all experience their issues on the same day, making it difficult to forecast the daily workload for the Detractor Squad.

The optimal workload for the squad corresponds to a specific number of calls handled per day, referred to as the operational target of daily calls. Achieving this target requires balancing the risks of overestimating or underestimating the workload, which will be addressed later.

To address this difficulty of balancing workload and determining when clients should be redirected to the Detractor Squad, NOS implemented a two-stage approach. The first stage involved the use of a classification model to assign each client a daily score between 0 and 1, reflecting the likelihood of them calling within the next 15 days. These scores were compiled into a list, with clients being prioritized based on their score. Once the likelihood scores were calculated, a time series model was used to forecast the optimal number of clients to flag each day from the ordered list. A flagged client is one identified as having a high probability of making a call, allowing their calls to be automatically prioritized and redirected to the Detractor Squad for expedited resolution. Since flagged clients might not place their calls on the exact day predicted, the optimal number of clients to flag each day was not fixed and did not directly correspond to the operational target for daily calls. Such variability had to be accounted for by the model. Each forecast was conducted two days in advance to allow for potential operational adjustments based on the prediction.

By combining the outputs of the classification and the time series forecasting model, NOS sought to direct the appropriate number of calls to the Detractor Squad, ensuring more effective intervention, resource allocation, and ultimately, increased operational efficiency. Accurately estimating the volume of clients to redirect was crucial to ensure the optimal utilization of the Detractor Squad. Both overestimation and underestimation could lead to inefficiencies. Overestimating the number of clients to flag can lead to redirecting calls that do not need the Detractor Squad's attention. These calls occupy slots that could otherwise be allocated to calls that truly require intervention. An underestimation, on the other hand, results in the Detractor Squad having insufficient calls to handle, leading to periods of inactivity. These idle moments are a financial burden, as the team incurs costs without adding value. Since the cost of idle time in the Detractor Squad is higher than the potential cost of clients needing to call again because they were not properly advised initially, overestimation is preferred.

1.3 Dataset

The work presented in this thesis, comprising the group section that describes the work done within the Business Analytics Masters' Project-Based Learning program, which will be further introduced in section 1.5 (Project-Based Learning Project), and the four individual parts rely on historical, pseudo-anonymized data from NOS' call center, covering the period from May 31, 2023, to September 29, 2023. Throughout this timeframe, over 2 million calls were treated by NOS' call center with more than 1.5 million clients.

Almost 20% of all the clients were relapsing at least once during the period, highlighting the business challenge at hand. From a model performance perspective, the existing predictive model did not fully meet its objectives, directing fewer than the target established for the Detractor Squad. This discrepancy indicates that, on average, the Detractor Squad has the capacity to handle a greater volume of calls, highlighting the need for model optimization.

One key variable in this dataset for this task is the aforementioned score for each client, reflecting the likelihood of a client calling within the next 15 days and generated by NOS' existing classification model. Additionally, information on whether the client made a call on the preceding day is included, which is crucial for defining the target variable, as explained further in section 1.5.1 (Target Variable and Data Preprocessing). Furthermore, this data captures various aspects of client interactions and service usage, along with past call center statistics, which play an important role in the multivariate models used in this project as described in section 1.5.3.2.4 (Multivariate Models) and in feature engineering discussed in section 1.5.2 (Feature Engineering).

In terms of client interactions, the dataset provides detailed records of calls answered by the call center, including attributes such as the teams handling the calls, call duration, hold times, and the number of transfers involved. Regarding service usage it contains information about the outcomes of house visits by service professionals, including metrics such as wait times for service and the occurrence of the visits. The past call center statistics are organized across multiple time intervals, ranging from a few days to multiple years. These statistics encompass

a variety of metrics, such as the number of calls made during each time frame, the total time spent on calls, total hold time, and a heuristic score that assigns greater weight to more recent calls.

1.4 Literature Review

This section provides a comprehensive overview of the methods, models, and advancements in time series forecasting relevant to this study. Starting with a general exploration of time series forecasting, it examines the foundational theories, statistical models, and machine learning techniques that have shaped this field. Specific attention is then given to the application of time series forecasting in call centers, highlighting the models used to predict call volumes and optimize operations. Finally, the section discusses the challenges of forecasting with small datasets and examines why neural networks may not be suitable in such contexts, setting the foundation for the methodological choices made in this study.

1.4.1 Time Series Forecasting

To effectively manage the dynamic nature of relapsing callers and accurately predict daily call volumes, time series forecasting becomes an essential tool. By analyzing historical data patterns, these methods can assist in anticipating future call volumes (Bastianin, Galeotti, and Manera 2019), thereby optimizing call center operations within the Detractor Squad.

A time series is a sequence of data points recorded at successive points in time (Chatfield 2000). The observations are collected over intervals of time scales such as seconds, minutes, hours, days, months, or years. Each observation y_t represents a value measured at a specific time t and the series $(y_1, y_2, \dots, y_{t_{\max}})$ captures how these values evolve over time (Hamilton 1994).

The task of time series forecasting is then defined as the attempt to predict data points in this series that are not yet observed, y_s , with $s > t_{\max}$, by leveraging statistical properties present in the historical data up to time $t_{\max}$. These methods rely on historical values of the series, also known as lagged values, and may incorporate external factors or explanatory variables to improve accuracy (Chatfield 2000). The ultimate goal is to analyze the underlying structure of

the data, such as trends, seasonality, cyclicity, and irregular fluctuations, to generate meaningful forecasts (Koopman and Lee 2009).

Trends represent long-term directional movements in the data, such as persistent increases or decreases over time (Wu and Zhao 2007). Seasonality captures recurring patterns at fixed intervals, such as daily, monthly, or yearly variations, while cyclicity reflects longer-term oscillations possibly driven by factors like economic or environmental cycles (Davey and Flores 1993). Irregular fluctuations, on the other hand, account for random, unpredictable variations that cannot be explained by these systematic components (K. Zhang, Ng, and Na 2020). These principles make time series forecasting applicable in fields like finance (e.g., stock prices), economics (e.g., GDP growth rates), and meteorology (e.g., temperature records), where the primary goal is to uncover and understand the underlying structure of the data (Yildiz and Kundakcioglu 2024).

Time series forecasting is a well-established field with a long history of development. Numerous methods and algorithms have been developed to address the task of time series forecasting. However, no single model can consistently provide the best forecasting results for all types of time series data due to the inherent complexity of the data. Additionally, the diverse range of use cases across different fields introduces unique characteristics and challenges that further complicate the forecasting process (Allende and Valle 2016).

Driven by advances in both statistical methods and computational power, the field of time series forecasting has evolved significantly over the past century. One of the first milestones was achieved when Yule 1927 introduced the concept of stochasticity in time series. Building on this foundation, researchers such as Slutsky, Walker, Yaglom, and Yule formulated the concepts of autoregressive (AR) and moving average (MA) models, which enabled the modeling of dependencies in sequential data (De Gooijer and Rob J. Hyndman 2006).

In the 1950s and 1960s, exponential smoothing methods emerged as a popular approach for time series forecasting. These methods, introduced by Brown 1959, Healy and Brown 1964, Holt 2004, and Winters 1960, gained popularity for their simplicity and effectiveness in handling trends and seasonality. Exponential smoothing assigns exponentially decreasing weights to past

observations, emphasizing more recent data, and is often used to complement other forecasting techniques (Brown 1959, Healy and Brown 1964, Holt 2004, Winters 1960).

In 1970, Box and Jenkins developed the Autoregressive Integrated Moving Average (ARIMA) model, marking a significant breakthrough in time series forecasting. ARIMA provided a framework for modeling non-stationary time series through differencing and the combination of AR and MA components (Box et al. 2016). This model specifically addresses the issue of non-stationarity by incorporating a differencing term, which represents the number of differencing operations required to achieve stationarity, thereby simplifying the complexities within time series data (Shumway and Stoffer 2017). To specifically address seasonal variations, the Seasonal Autoregressive Integrated Moving Average (SARIMA) model was subsequently developed, extending the ARIMA framework by incorporating seasonal differencing as well as seasonal autoregressive and seasonal moving average components (Box et al. 2016, Chapter 9).

Building on the foundational advancements of ARIMA and SARIMA, the integration of external information further enhanced the flexibility of time series models. The development of the Seasonal Autoregressive Integrated Moving Average with Exogenous Variables (SARIMAX) model extended the SARIMA framework by allowing the inclusion of external factors that influence the time series (Korstanje 2021a). By accounting for these exogenous influences, SARIMAX provided a versatile tool for handling complex, real-world data. Vagropoulos et al. 2016 demonstrated that SARIMAX can outperform SARIMA in day-ahead forecasting, showcasing its competitiveness with other established methods.

The late 20th century saw the emergence of machine learning models for time series forecasting, introducing a new paradigm. Ensemble methods gained prominence due to their ability to model non-linear relationships (Connor, Martin, and Atlas 1994, Chakraborty et al. 1992). They combine the predictions of multiple weak learners to create a more accurate model (Thomas G Dietterich 2000). Extreme Gradient Boosting (XGBoost), introduced by Chen and Guestrin 2016 is one example of this class of models, which iteratively refines predictions through weighted corrections of prior errors. The incorporation of regularization mechanisms helps mitigate overfitting, making it particularly effective in complex forecasting tasks (Bühlmann 2012).

In parallel, advancements in deep learning have further expanded the forecasting toolkit. Long Short-Term Memory (LSTM) networks, introduced by Hochreiter and Schmidhuber 1997, addressed key limitations of traditional neural networks in handling sequential data. By capturing long-term dependencies in time series, LSTM has become a cornerstone of modern deep learning applications in forecasting. These models have significantly advanced the field, enabling robust handling of complex temporal patterns and non-linear relationships (LeCun, Bengio, and Hinton 2015).

In the early 2000s, the Theta model gained recognition for its simplicity and robustness (Vassilis Assimakopoulos and Nikolopoulos 2000). This model was particularly highlighted during the M3-competition, a renowned forecasting competition aimed at evaluating the accuracy and utility of various time series forecasting methods (De Gooijer and Rob J. Hyndman 2006, Chapter 2.2). This model decomposes de-seasonalized data into two "Theta lines": the first emphasizes the long-term trend by completely smoothing the curvature, while the second amplifies curvature to capture short-term dynamics. These components are recombined to create a robust forecast that balances long-term and short-term patterns effectively. The model is particularly notable for its simplicity and adaptability across different time series characteristics (Vassilis Assimakopoulos and Nikolopoulos 2000).

More recently, in 2017, Facebook's Data Science Team introduced the Prophet model (S. J. Taylor and Letham 2018). Designed for practical use, Prophet employs a decomposable framework incorporating trend, seasonality, and holidays, making it particularly effective for time series with irregularities such as missing data or outliers (Vishwas and Patel 2020). With its intuitive interface and flexibility, Prophet allows users to adjust parameters without requiring an extensive understanding of the underlying model (S. J. Taylor and Letham 2018). Its multivariate capabilities also enable the simultaneous analysis of multiple variables (F. Wang and Aviles 2023) and the incorporation of exogenous variables (Shumway and Stoffer 2000), enhancing its predictive power in complex scenarios (R. Hyndman and Athanasopoulos 2012).

1.4.2 Time Series Forecasting at Call Centers

Given its versatility, time series forecasting has been extensively applied to predict call center arrivals, where accurate predictions are vital for optimizing staffing levels and managing resources efficiently. Early studies primarily employed classical statistical models, such as SARIMA and exponential smoothing methods, which can capture basic trends and seasonality in call arrival data (J. W. Taylor 2007, Petropoulos, Apiletti, et al. 2022, Chapter 3.8.4.). However, the evolution of forecasting mirrors broader trends in the field, shifting towards more advanced machine learning techniques. Particularly, deep learning models, such as LSTM and its variants have shown promise in handling large-scale call center datasets and complex temporal dependencies, providing more robust predictions (Torres et al. 2021, Lim and Zohren 2021).

1.4.3 Time Series Forecasting in Limited Size Datasets

In both the general introduction of time series forecasting and the more specific discussion on its application to predicting call center arrivals, a clear trend emerges: the increasing reliance on neural networks and deep learning models.

As highlighted by LeCun, Bengio, and Hinton 2015, neural networks, however, require large datasets to train effectively because they need a substantial amount of data to properly adjust their numerous parameters and avoid overfitting. “Overfitting occurs when the gap between the training error and test error is too large” (Goodfellow, Bengio, and Courville 2016, Chapter 5, p. 110) implying that the model learns not only the underlying patterns in the data but also the noise, which leads to poor generalization to new, unseen data. In the case of neural networks, this issue is particularly pronounced because these models have a large number of parameters that need to be adjusted. Moreover, pretrained neural networks, despite containing a significant number of non-learning neurons, face similar challenges, as they can still struggle with overfitting when insufficient data is provided.

1.5 Project-Based Learning Project

To address the challenge of predicting the optimal volume of customers to flag each day for the Detractor Squad as introduced in 1.2 (Context), the Project-Based Learning (PBL) program was undertaken. PBL is a pedagogical approach in which students are actively engaged in real-world projects, applying their knowledge and skills to solve concrete problems. It offers a hands-on learning experience that bridges the gap between theory and practice. Through the PBL program, the opportunity was provided to delve deeper into the second stage of NOS' two-stage approach: the time series forecasting model used to determine the daily number of customers to flag. Specifically, efforts were focused creating a model that outperforms the ARIMA (1,0,1) model currently implemented by NOS for this purpose, on the time series data at hand. This section presents a detailed overview of the PBL project and its outcomes.

1.5.1 Target Variable and Data Preprocessing

As outlined in section 1.2 (Context), the objective of the PBL project was to predict the optimal volume of customers to flag each day for the Detractor Squad. However, the dataset, as detailed in section 1.3 (Dataset), was structured at the client level rather than daily granularity, resulting in the absence of a suitable target variable. Consequently, the initial step within the PBL program involved the transformation of the dataset into daily granularity to establish an appropriate target variable. Considering that the Detractor Squad exclusively manages inbound calls, which are calls initiated by clients, the dataset was filtered to eliminate any records of outbound calls, i.e. calls initiated by NOS' call center. Furthermore, since the Detractor Squad does not operate on weekends, any records from weekends were also excluded from further analyses. The resulting target variable, representing the number of people to flag in order to meet the daily target of actual calls for the Detractor Squad, is shown in figure 1.

The x-axis represents the time period from May 31st to September 29th, 2023, while the y-axis shows the number of clients flagged. From the start of the dataset until late July, the figure displays a spiky pattern. Notably, during this period, clear seasonality was observed since spikes predominantly occurred on Mondays. Nevertheless, the final spike from that period deviated

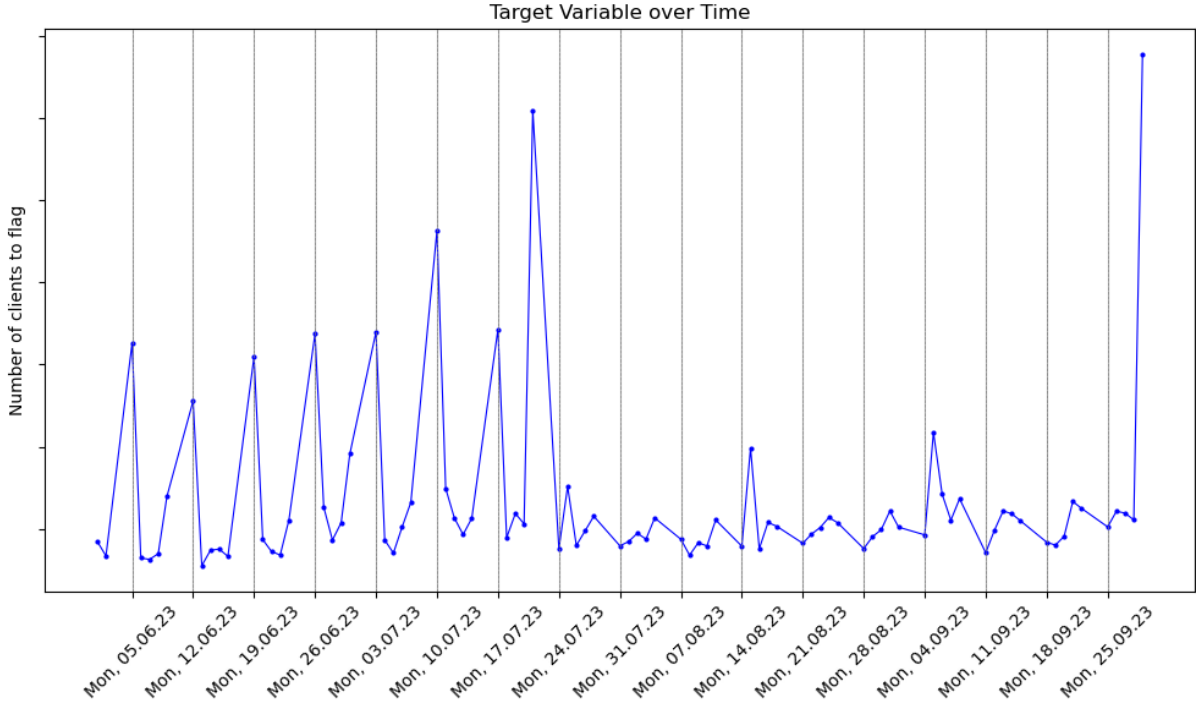


Figure 1: The target variable [#clients to flag] plotted over time

from this pattern by occurring on a Tuesday. Through close collaboration with NOS, these extreme spikes were identified as outliers in the target variable.

Following the initial spiky phase, a more stable pattern emerged from late July through the end of September. This stability, punctuated by only a couple of smaller spikes, all occurring on a Tuesday, was, according to NOS, more representative of the overall behavior of the data. A notable feature of this dataset was the massive spike in the target variable at the end of the provided time series. This abrupt increase could signify a seasonal shift, a change in client behavior, or an external event impacting the number of clients needing attention. Similar to the extreme spikes observed during the previously mentioned spiky period, this spike was considered an outlier, confirmed by NOS.

1.5.2 Feature Engineering

Feature engineering was conducted following the same logic as the computation of the target variable, with daily-grouped features being created from the original, more granular data. Since the objective was to predict the number of clients to flag each day, variables that could impact call volume and behavior were prioritized. The business expertise provided by NOS was

invaluable in guiding decisions regarding the selection of variables.

The features deemed most important included call duration, hold time, total number of calls to the technical line, and client score. Call duration referred to the length of each interaction, while hold time indicated the duration clients spend waiting for assistance. The client score, as already outlined in section 1.3 (Dataset), reflected the likelihood of a client calling within the next 15 days. Additionally, information extracted from the date, such as the weekday and month, were also considered significant.

Importantly, rather than relying on a single metric for these variables, their daily sum, mean, and standard deviation were employed, providing a more comprehensive representation of daily patterns in the dataset.

1.5.3 Methodology

1.5.3.1 Target Capping

As mentioned in section 1.5.1 (Target Variable and Data Preprocessing) and evident in figure 1, the target variable exhibits both numerous and extreme outliers. This resulted in a spiky pattern from the start of the dataset until late July, along with a significant spike at the end of the dataset. The high quantity of extreme outliers was observed to distort the model's understanding of typical patterns, leading to inaccurate predictions and an inability to generalize effectively to the underlying data distribution. Consequently, strategies such as capping the target variable, were implemented to mitigate the adverse effects of outliers and enhance overall predictive accuracy.

Capping started by identifying these extreme values using an Interquartile Range (IQR) technique (Tukey 1977) with a multiplier of 1.5. The IQR represents the central 50% of the data, i.e. the range between the first and third quantiles. Any point outside 1.5 times the IQR above the third quantile or below the first quantile was considered as an outlier.

After identifying the outliers, these extreme values were adjusted to the maximum value in the non-outlier data. However, the difference between the original outliers and the capped values

was not entirely discarded. Instead, the difference was scaled down by 20%, preserving some of the relative size of these spikes to ensure that the pattern of variability in the data was maintained while diminishing their impact.

The chosen hyperparameters for IQR multiplication and scaled difference resulted from simple experimentation with various combinations to identify the optimal configuration. Figure 2 illustrates the impact of capping on the operational target. The blue line represents the original operational target, whereas the red line indicates the capped version. The capping effectively reduced the magnitude of extreme spikes while preserving the overall structure and trends of the data, ensuring a more stable and practical input for forecasting models.

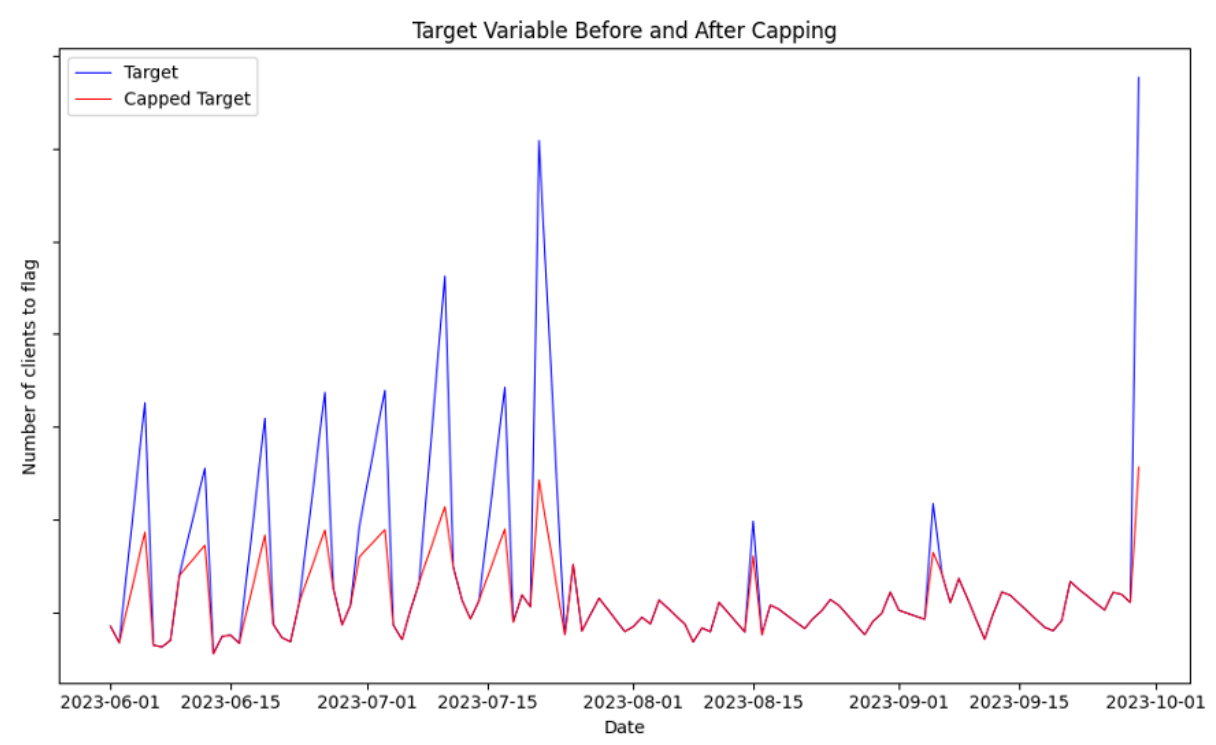


Figure 2: Impact of the capping on the target variable

1.5.3.2 Introduction of the Models Deployed

During the project, a structured three-step approach was employed, progressing from baseline models to univariate models, and finally to multivariate models. This systematic progression enabled an incremental increase in complexity.

1.5.3.2.1 Limitations due to Small Dataset

As discussed in section 1.3 (Dataset) the time series available spanned from May 31, 2023, to September 29, 2023, covering only 122 days. Once weekends were excluded, as mentioned in section 1.5.1 (Target Variable and Data Preprocessing), the dataset was further reduced to 88 days. This posed a challenge for the analysis since time series are characterized by recurring seasonal trends (Rob J. Hyndman and Athanasopoulos 2018). With only four months of data and given the daily granularity, complete seasonality was absent, and the diversity of observations was limited, making it difficult to capture these trends and generate accurate future predictions. Additionally, the small dataset increased the model's sensitivity to outliers, complicating the distinction between significant signals and noise. This further encouraged overfitting, which compromised the model's ability to generalize to new data (Shumway and Stoffer 2000).

Moreover, as highlighted in section 1.4.3 (Time Series Forecasting in Limited Size Datasets), neural networks, while powerful, require large datasets to avoid issues such as overfitting, where the model learns the noise rather than the underlying patterns. Given these constraints and the risk of poor generalization with small datasets, the focus during the PBL program was not placed on complex neural networks but rather on simpler models with fewer, more easily tunable parameters. This approach was adopted to mitigate the risks associated with limited data and to develop models that were more stable and reliable for the analysis.

1.5.3.2.2 Baseline Models

The initial focus was placed on establishing baseline models that served as an essential point of reference for the performance of subsequent, more sophisticated models. The first baseline model predicted the next value by using the average of data from a fixed window of past time points.

As discussed in section 1.5.1 (Target Variable and Data Preprocessing), consistent patterns in the data corresponding to specific weekdays were identified. This observation led to the implementation of the Weekday Average model, which was designed to capture these recurring

patterns. While effective at highlighting significant peaks, this model also revealed the challenges posed by abrupt changes in the data, such as the spiky pattern from late May to late July, followed by a period of stability extending through September. The model’s sensitivity to such fluctuations emphasized the need to evaluate the stability and consistency of predictions, in addition to traditional error-based metrics. This dual focus on correctness and consistency, which is further explored in section 1.5.3.3 (Model Evaluation Criteria) became a critical aspect of the modeling approach.

In addition to these models, ARIMA (1,0,1) was incorporated, recognizing its significance as the model currently employed by NOS. By comparing the results with this model, potential advancements in the performance of the attempted approaches could be highlighted. Thus, this model served as the most important benchmark throughout the project and was referred to as the “NOS Benchmark” throughout the project.

Throughout the baseline phase, an embed size of 30 was consistently utilized, aligning with the configuration currently used by NOS. This embed size defined the maximum training window for each iteration of the temporal cross-validation (TCV), where the training set iteratively grew each day until it reached the predetermined embed size and then remained fixed. A more detailed explanation of the TCV approach is provided in section 1.5.3.3.4 (Temporal Cross Validation).

1.5.3.2.3 Univariate Models

In the second phase, univariate time series analysis was conducted, and several prominent models were applied to capture different patterns and trends in the data.

The univariate analysis was initiated with the ARIMA model, as this model specifically addressed the issue of non-stationarity by incorporating a differencing term, as mentioned in section 1.4.1 (Time Series Forecasting). Given the pronounced fluctuations and the absence of a constant mean or variance in the dataset, the differencing component was pivotal in addressing these irregularities.

The limitation of having only 88 days of data, constrained the ability to identify long-term trends and seasonal effects. Despite this limitation, ARIMA models are generally still effective in capturing short-term dependencies (Shmueli and Polak 2024, Box et al. 2016) aligning with the objective of predicting two workdays ahead.

While ARIMA is generally effective for non-seasonal data, the SARIMA model was adopted to enhance the analysis due to the periodic patterns observed in the dataset, outlined in section 1.5.1 (Target Variable and Data Preprocessing). By addressing these periodic patterns, SARIMA proved valuable for datasets with recurring fluctuations (Box et al. 2016), such as the ones at hand. The need to account for the cyclical behaviors, which motivated the introduction of the Weekday Average model, as discussed in section 1.5.3.2.2 (Baseline Models), highlighted the importance of accurately modeling consistent patterns over time. This approach allowed seasonality to be explicitly accounted for and the periodic patterns, critical for accurately modeling consistent behaviors, to be captured.

To complement the SARIMA model, Exponential Smoothing was also applied, leveraging its adaptability to respond effectively to shifts in data patterns. This approach ensured sensitivity to short-term changes in the time series, enhancing forecast accuracy.

The last univariate model that was implemented was the Theta model, due to its adaptability across different time series characteristics (Vassilis Assimakopoulos and Nikolopoulos 2000). Notably, this model was also employed at NOS prior to their adoption of the ARIMA (1,0,1) model, thereby providing a compelling rationale for its implementation in this comparative analysis.

1.5.3.2.4 Multivariate Models

Besides tackling univariate modeling, the project also involved exploring multivariate models to capture potential relationships within the data that simpler univariate models could have missed.

The first multivariate model deployed was a multivariate extension of the SARIMA model, SARIMAX, as it is a natural progression from the univariate econometric models ARIMA and

SARIMA, discussed in the previous section. Furthermore, this model offers greater flexibility when dealing with complex time series data, which was particularly relevant given the nature of the time series (Korstanje 2021a).

Following SARIMAX and continuing along the multivariate time series modeling trajectory, Prophet was selected due to its capability to handle data with various components such as trend, seasonality, and holiday effects (S. J. Taylor and Letham 2018). This feature is particularly advantageous given the cyclical nature of the dataset. Additionally, the model is known for efficiently handling outliers and offering fast model fitting (Vishwas and Patel 2020), making it a suitable choice for addressing the spiky patterns present in the data and the need for daily forecasts.

After exploring time series models such as SARIMAX and Prophet, a transition was made to machine learning approaches to explore an alternative technique and potentially achieve improved results. The final multivariate model considered was XGBoost, chosen for its ability to capture the complexity of the data (Yan Wang and Guo 2020). Moreover, it effectively mitigates overfitting through regularization (Chen and Guestrin 2016), addressing the generalization preferences highlighted by NOS.

1.5.3.3 Model Evaluation Criteria

An inappropriate choice of evaluation metrics can adversely impact the entire model lifecycle, from development to deployment. As mentioned in section 1.5.3.2.2 (Baseline Models), the importance of a dual approach that balanced both performance and consistency was demanded by NOS. From a performance perspective, the model was required to accurately predict the daily call volume for the Detractor Squad, ensuring that relapsing clients received adequate attention. On the consistency side, the model needed to deliver stable predictions that minimized day-to-day fluctuations in call volume, supporting a balanced workload for the Detractor Squad and ensuring operational efficiency. This dual focus ensured that both the immediate demands of relapsing clients and the long-term sustainability of team performance were addressed.

1.5.3.3.1 Measurement Variable

The target variable, which represented the number of clients to flag each day, did not align well with the operational reality of the Detractor Squad. Since their workload was determined by the number of calls they handled daily, using flagged clients also as a measurement variable, i.e., the variable on which the evaluation metrics are based, became impractical. Given that not all flagged clients would initiate calls, relying on flagged client predictions led to significant discrepancies in the resultant call volume. A model might indicate a need to flag many clients, but this does not necessarily correlate to a similar increase in the actual number of calls received. This inconsistency complicated the evaluation of the model's true effect on the Detractor Squad. To address this, a new measurement variable was introduced, representing the number of daily calls resulting from flagging a specific number of clients. To determine the number of clients that should have been flagged to align with operational requirements, the daily ordered list of client scores was used, along with the column indicating whether each client made a call on that specific day. The list of clients who made a call was then progressed through until the desired call volume was reached, thereby meeting the operational needs. The same approach was applied to calculate how many calls would have resulted from the model's predictions. This adjustment provided a more relevant and practical assessment of the model's performance, as it allowed for direct comparison with the operational call volume requirements.

It is important to clarify that the target variable remained the number of clients to flag each day. The daily call volume requirement for the Detractor Squad solely served as a measurement variable for evaluating the model's performance. Therefore, while the models were required to predict the number of clients to flag, the error and smoothness metrics were calculated based on the actual number of calls received, reflecting a more meaningful assessment of model performance. These metrics will be explored in the subsequent section.

1.5.3.3.2 Evaluation Metrics

To evaluate the models, two distinct metrics were employed in line with the dual approach outlined in section 1.5.3.3 (Model Evaluation Criteria). To assess performance, the Mean Ab-

solute Percentage Error (MAPE) (Korstanje 2021b) was selected. This metric was chosen due to its ability to present errors in relative terms rather than absolute values, enabling consistency across varying data scales and ensuring reliable comparative evaluations. Moreover, MAPE was already in use by NOS for this particular problem, aligning the approach with existing practices.

The Smoothness Index (SI) was selected as the metric to assess stability. Defined as the Mean Squared Differences divided by the Variance of the observations, the SI measures the variability between predictions. This metric, developed by the PBL team and validated in collaboration with NOS, was used to track the model's stability and consistency throughout the forecasting process. Monitoring the SI ensured that the Detractor Squad's workload remained relatively consistent, minimizing significant fluctuations in day-to-day operations.

1.5.3.3.3 Best Case Scenario

As discussed in section 1.2 (Context), overestimating call volumes was prioritized by NOS rather than underestimating them, given the higher cost associated with idle time in the Detractor Squad. To reflect this preference, a tolerance of 50 calls above the daily operational target was established in collaboration with NOS, creating a target-tolerance band, as shown in figure 3. Any forecast that fell within this specified range was deemed to be ideal, representing what was referred to as the “best-case scenario”. If forecasts fell just below the target by one call or exceeded the upper tolerance threshold by one call, both instances were considered equally distant from the best-case scenario, underscoring the emphasis on overestimation.

This approach not only aligned with the operational preferences of NOS but also provided a structured framework for evaluating the performance of the forecasting models. By establishing a clear definition of what constituted an optimal outcome, the degree to which each developed model approached the best case scenario could be assessed. Consequently, the results presented in this study consistently indicate the degree to which each model deviated from the “best-case scenario”.

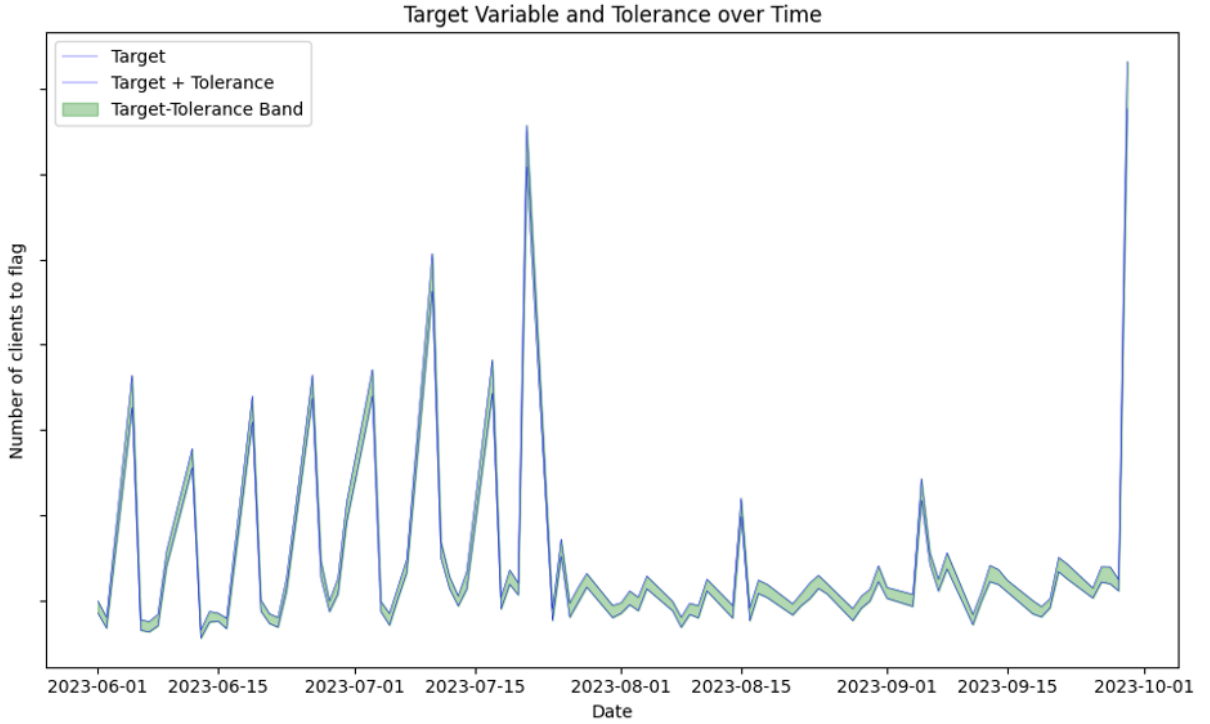


Figure 3: Target-Tolerance Band

1.5.3.3.4 Temporal Cross Validation

To evaluate the effectiveness of a predictive model, it was crucial to ensure that the model performed well not only on the data used for training but also on new, unseen data. To achieve this, a common approach involved splitting the available data into separate subsets, allowing the model to be trained on one part while being tested on another. This idea formed the basis of what is known as cross-validation (Berrar et al. 2019), a technique where the data is divided into multiple subsets, or folds, and the model is trained and tested across different combinations of these subsets.

In time series data, cross-validation is not appropriate because the data has a natural time order (Cerqueira, Torgo, and Mozetič 2020, Bergmeir and Benítez 2012, Falessi et al. 2020). Randomly splitting the data for cross-validation could produce misleading results, as training on future data and testing on past data could lead to data leakage and compromise the evaluation's validity. Instead, techniques like forward validation, introduced by Hjorth 1994, were used to respect the time order by only using past data to predict future outcomes.

The core concept of these techniques revolve around the rolling window approach. This method begins by selecting an initial subset of data and then progressively shifts the window forward with each prediction iteration. It is crucial to note that a predetermined embed size, established as a hyperparameter, must be defined before implementing this method (Bergmeir and Benítez 2012).

By training on successive subsets of data rather than the entire dataset, the model was able to adapt to evolving temporal patterns, hence mitigating the risk of overfitting (Berrar et al. 2019), a significant concern given the size of the dataset.

Therefore, to validate the models, temporal cross-validation (TCV) was developed, drawing inspiration from forward validation. In TCV, an initial split was made where the training set comprised data from one day, and the test set consisted of data from two days ahead, aligning with NOS' request for predictions to be made two days in advance, as explained in section 1.2 (Context).

As the process progressed, each new split incorporated the previous test set into the training set, which then expanded sequentially by one day. Once the training set reached the predetermined embed size, the process continued by rolling the window forward. A visualization of the TVC, given an embed size of 15 is shown in figure 4 below.

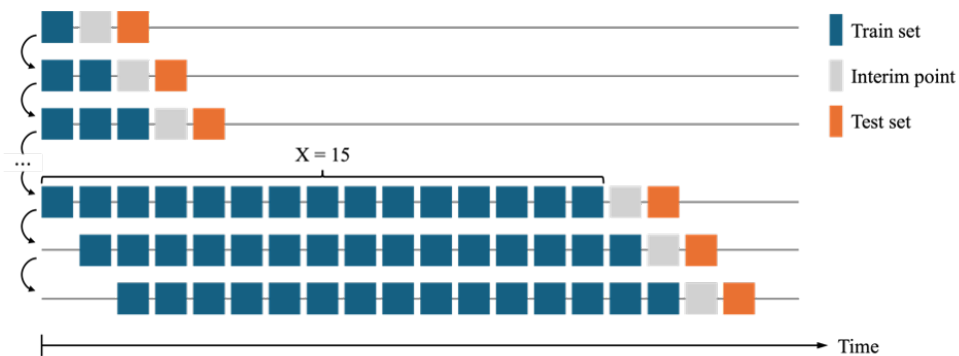


Figure 4: Visualization of the Temporal Cross Validation given an embed size of 15

1.5.3.3.5 Evaluation Window

The implementation of temporal cross validation introduced a new challenge. While optimal performance was achieved by various models with different embed sizes, this variation in window size led to an unequal number of predicted data points across models. While various models may achieve optimal performance with different embed sizes, this variation in embed size could lead to an unequal number of predicted data points across models. Consequently, when comparing model performance, this discrepancy could introduce bias, potentially skewing the evaluation results.

To address this issue, model performance was tracked for the last 40 days based on two key factors. First, all models were developed using an embedding size of 30 days or less due to the dataset's limited size, which prevented capturing meaningful seasonality patterns with larger embedding sizes. Second, the initial period from May to July exhibited a spiky pattern, as noted in section 1.5.1 (Target Variable and Data Preprocessing), while the last 40 days provided a more representative view of typical operational patterns, a characteristic confirmed by NOS.

Furthermore, the final day was excluded from the metrics analysis, as it was considered an outlier that could have disproportionately affected the metrics and led to misleading results, as discussed in section 1.5.3.1 (Target Capping). This approach ensured that all models were evaluated on 39 predictions, allowing for fair and unbiased comparisons while mitigating the impact of outliers.

1.5.3.3.6 Tailored hyperparameter tuning

To optimize the performance of the models, grid search was employed, a hyperparameter tuning method widely used in data science. This approach exhaustively explored a pre-defined subset of hyperparameter combinations for the model it was applied to. By evaluating every possible configuration, grid search determined the optimal parameters that delivered the best results according to a specific metric.

In this case project, a customized version of grid search was required that could first accommo-

date variable parameters and embed sizes, and secondly, allow comparisons of different metrics, aligning with the aforementioned dual approach. As mentioned in the previous section, outliers were consistently excluded from the metric calculations to ensure the integrity of the MAPE results.

In conjunction with grid search, the multi-objective optimization problem was tackled by employing the concept of Pareto Efficiency, named after the Italian economist Vilfredo Pareto (Ehrgott 2012, Pindyck et al. 2018, Chapter 16.2). A model's outcome is Pareto efficient if any possible reconfiguration of the hyperparameters would lead to a deterioration in at least one metric (Yao et al. 2023). In this context, reducing MAPE required adjustments of the hyperparameter configuration, which, in turn, would have led to a less favorable SI, and vice versa. All these Pareto efficient configurations collectively formed the so-called Pareto Frontier. By focusing on this frontier, non-Pareto efficient points that did not contribute to the goal of achieving an optimal balance between MAPE and SI were disregarded.

This approach enabled the identification of the most promising hyperparameter configurations while also providing a deeper understanding of the trade-offs inherent in the model's performance.

1.5.4 Results from PBL

In this analysis, the performance of the forecasting models was evaluated by comparing them to the established benchmark, providing a clear assessment of their effectiveness and potential impact on NOS' operations. To guide model selection, Pareto frontiers were created for each model, highlighting configurations that achieved the optimal balance between MAPE and SI.

Figure 5 provides a visual comparison of all algorithms and their respective tested configurations through the application of the Pareto-frontier concept, as introduced in section 1.5.3.3.6 (Tailored hyperparameter tuning). Figure 5a illustrates the Pareto-efficient configurations within each algorithm, while figure 5b presents the overall Pareto-frontier (OPF), indicated by the black dashed line, encompassing all configurations tested. In both panels, each algorithm was distinguished by a unique color. To indicate whether configurations incorporated the target capping

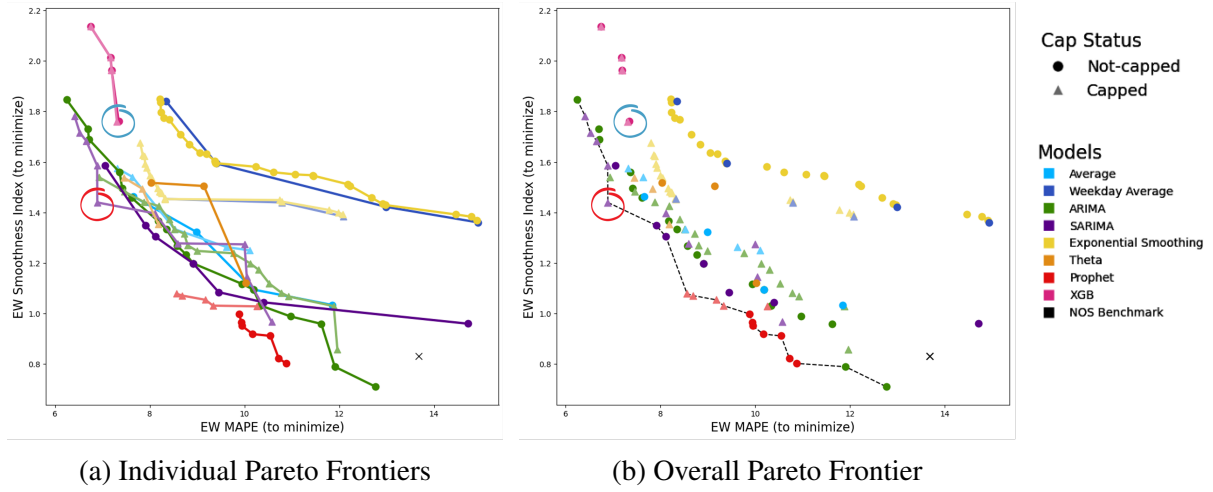


Figure 5: Pareto frontier PBL

concept, introduced in section 1.5.3.1 (Target Capping), different marker shapes were employed where circles denoted configurations based on non-capped data, whereas triangles represented configurations based on capped data.

In general, figure 5 demonstrated that increased stability often came at the expense of predictive power. Along the OPF, starting from the top, econometric univariate models such as ARIMA and SARIMA dominated, indicating that these models performed well in terms of forecast precision, achieving lower MAPE values, but exhibited high instability. Further down the OPF, the SARIMA model continued to appear, reflecting its ability to achieve strong results with low MAPE values while outperforming the majority of other algorithms in terms of stability with lower SI values. Toward the lower end of the OPF, Prophet models were followed by ARIMA models, which demonstrated the best performance in terms of stability but tended to sacrifice precision, as reflected in higher MAPE values.

Notably, the NOS Benchmark model, represented by the black x and positioned in the low SI and high MAPE region, did not appear on the OPF. This absence suggested that, compared to the other models, it failed to achieve the optimal balance between MAPE and SI. Consequently, selecting any model from the OPF offered a more effective solution for the forecasting task

related to call center arrival predictions than the benchmark currently utilized by NOS.

1.5.4.1 Deployable Solution

In the pursuit of identifying the most practical model for deployment, it was essential to select a configuration that achieved the optimal balance between MAPE and SI. The SARIMA $[(2,0,1),(0,0,1,20)]$ configuration with a 20-day embed size, combined with the target capping method introduced in section 1.5.3.1 (Target Capping), is illustrated by the red-outlined point in figure 5. From here on referred to as the deployable solution, this configuration delivered a MAPE of 6.89 with an SI of 1.44. Compared to the NOS Benchmark, with a MAPE of 13.68 and SI of 0.83, this represented a 49.6% reduction and a 73.5% increase, respectively. Although the increase in SI appeared considerable, the substantial reduction in error reflects a more favorable trade-off for the application at hand, prioritizing predictive accuracy over stability.

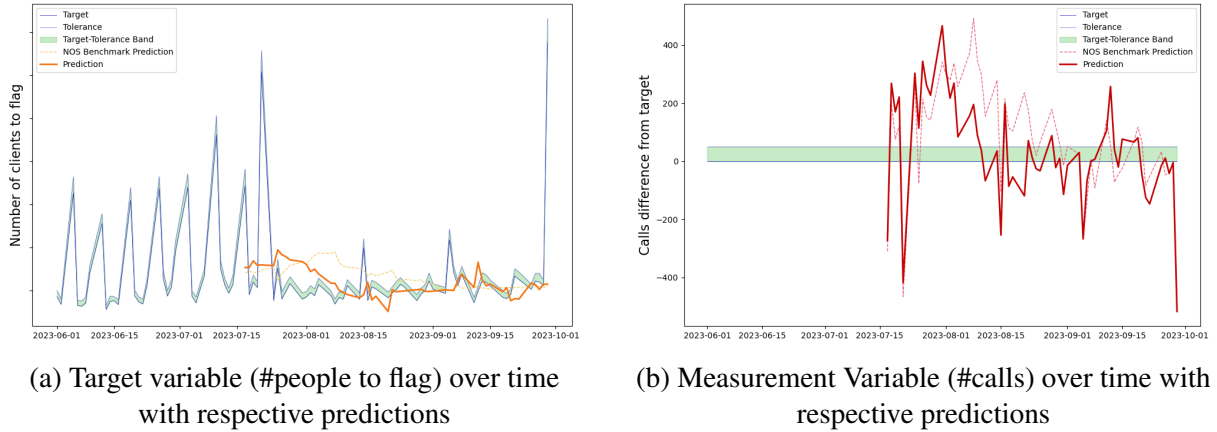


Figure 6: Performance of the Deployable Solution - PBL

Figure 6 visualizes the performance of the deployable solution. As established in section 1.5.1 (Target Variable and Data Preprocessing), the primary challenge for this forecasting task, given the dataset, was the abrupt transition in patterns from spiky to non-spiky.

Figure 6a illustrates the performance of the NOS Baseline model and the deployable solution on the target variable, defined in section 1.5.1 (Target Variable and Data Preprocessing) as the number of people to flag. During the transitional period of the time series, the deployable solution demonstrated a faster adaptation to the non-spiky pattern compared to the NOS Benchmark

model. Further, 6b, which illustrates the performance on the measurement variable, defined in section 1.5.3.3.1 (Measurement Variable) as the number of calls, shows that the deployable solution's predictions were closer to the target-tolerance of the measurement variable in most cases, which was quantified in the reduction in MAPE.

In conclusion, this SARIMA configuration provided a well-balanced trade-off between minimizing error and ensuring a manageable level of forecast variability.

1.5.4.2 Build-Onto Solution

As established, the deployable solution offered the best trade-off between MAPE and SI on the given dataset. However, the limitation regarding dataset size made a full optimization of the multivariate models infeasible. Therefore, due to the significance of this issue for call center operations, as discussed in section 1.2 (Context), a multivariate “build-onto” solution was provided alongside delivering the deployable solution, intended to serve as a foundation for further refinement and optimization if additional data becomes available.

Among the multivariate models evaluated, the XGBoost model demonstrated the lowest MAPE values observed throughout the project. The chosen configuration of the XGBoost algorithm is represented by the blue-outlined point in figure 5. This model employed 35 estimators, an absolute error regression objective, a maximum depth of 25, an L1 regularization term of 0.1, and a 10-day embed size. It achieved a MAPE value of 7.5, representing a reduction in error of more than 45% compared to the NOS Benchmark, highlighting its strong predictive accuracy. However, the XGBoost configuration recorded an SI value of 1.76, one of the worst performances regarding smoothness, indicating its inability to achieve a balance between accuracy and stability. This limitation also explained why it did not appear on the OPF.

Nevertheless, this model, being multivariate, can capture complex interactions in variables, as mentioned in section 1.5.3.2.4 (Multivariate Models). As such, it has much latent potential for further optimization and further exploration of performance with more datapoints, as well as with more aggregate variables from other sources.

2 Individual Part

Throughout the PBL project, the focus has been on analyzing and implementing a single forecasting model. In general, the rationale behind focusing on one single forecasting method is that there exists one “best” model that can be identified (Winkler and Makridakis 1983). Instead of relying on a single approach, one can train and combine multiple models through ensemble learning.

Herein, it is attempted to address the research question whether ensemble learning methods, including both homogeneous and heterogeneous models, can be utilized to improve forecasting accuracy and stability in the context of small datasets, particularly in comparison to single-model approaches.

Ensemble learning leverages the collective predictive power of multiple models, referred to as base-learners, to improve overall performance compared to individual models (Opitz and Maclin 1999). By aggregating the outputs of diverse base-learners, ensemble methods aim to reduce errors caused by individual model bias and variance (Sakib, Mustajab, and Alam 2024). According to Polikar 2006, ensemble methods have been capable of effectively addressing the issue of limited data.

This study focuses on two ensemble learning approaches, covering both homogeneous and heterogeneous methods. The first approach, bagging, is a homogeneous method derived from bootstrap aggregating (Breiman 1996), which consists of combining multiple instances of the same base-learner trained on different subsets of the data (Breiman 1996). The second approach, Dynamic Integration (DI), is a heterogeneous method that employs a diverse set of base-learners and dynamically selects or weights them based on their recent performance (Rooney et al. 2004). Three variants of DI were investigated, namely Dynamic Selection (DS), Dynamic Weighting (DW), and Dynamic Weighting with Selection (DWS).

To evaluate the effectiveness of these methods, this individual part builds on the PBL project, maintaining the objective of accurately estimating the volume of clients to redirect for optimal utilization of the Detractor Squad and employing the same dataset, characterized by limited

historical data. Six distinct statistical algorithms were employed, including ARIMA, SARIMA, Exponential Smoothing, and Theta, as well as simpler methods such as Average and Weekday Average. These algorithms formed the pool from which base learners were selected. Homogeneous configurations used only one algorithm, while heterogeneous configurations combined multiple algorithms, ensuring each algorithm was represented at most once.

This study is structured as follows: Section 2.1 (Literature Review) establishes the theoretical foundation by introducing ensemble learning, examining its effectiveness, and outlining its key characteristics. It further explores the use of ensemble methods in time series forecasting, focusing on bagging and DI. Section 2.2 (Methodology) details the methodological framework, describing the implementation of the two ensemble methods, with a focus on the choice of base-learners, ensuring diversity, and parameter optimization. Section 2.3 (Results & Discussion) assesses the performance of bagging and DI individually, culminating in a comparative analysis and discussion of the combined results. The discussion concludes with section 2.4 (Limitations & Future Work), which highlights the study’s limitations and outlines potential directions for future research, followed by section 2.5 (Conclusion), which summarizes the key findings, emphasizing the study’s contributions, practical implications, and recommendations for future application at NOS.

2.1 Literature Review

2.1.1 Introduction to Ensemble Methods

Ensemble methods involve combining the outputs of multiple individual base-learners (Yichang Wang 2021). By leveraging their combined outputs, the overall generalization ability of the ensemble can be significantly enhanced by smoothing out the noise of the individual base-learners, thus reducing the risk of overfitting (Thomas G Dietterich 2000).

While all ensemble methods share the common objective of combining multiple base-learners to improve accuracy and robustness, they differ significantly in their approaches and underlying principles (Zhou 2012). These differences can be categorized based on whether the method is homogeneous, which relies exclusively on a single type of base-learner, or heterogeneous,

which utilizes a variety of algorithms as base-learners (Zhou 2012). Additionally, the nature of how the base-learners interact distinguishes methods that employ parallel training, where the base-learners operate independently, and sequential training, where the base-learners build upon the results of previous base-learners (Thomas G Dietterich 2000). Regardless of the approach, the final step of the ensemble is to aggregate the predictions of the base-learners to produce a unified output. A few aggregation methods will be explored in section 2.1.4 (Aggregation Methods).

This vast array of options available for constructing ensemble methods contributes to the broad spectrum of ensemble learning approaches. Bagging, an abbreviation for bootstrap aggregating (Breiman 1996), and boosting (Schapire 1990) are two of the most prevalent ensemble learning methods, both of which are homogeneous approaches. While bagging trains in parallel, where each base-learner is trained on a different subset of the training data generated through bootstrapping (Effron and Tibshirani 1994), boosting constructs a sequence of base-learners, as mentioned in section 1.4.1 (Time Series Forecasting).

Since the foundational works of Breiman 1996, and Schapire 1990, the popularity for utilizing ensemble learning methods increased across various domains and applications, encompassing both classification and regression tasks, as well as time series forecasting (C. Zhang and Ma 2012, Sagi and Rokach 2018). While many ensemble learning methods were initially designed for classification tasks, leading to the term "multiple classifier systems" (Zhou 2021), the application of ensemble methods to time series forecasting predates the two seminal works introducing bagging and boosting. Specifically, Bates and Granger 2017 pioneered the use of an ensemble learning approach for time series forecasting by combining two forecasts on an airline passenger dataset.

In addition to bagging and boosting, there are many other ensemble methods available, such as stacking, blending, cascading, or voting, that will not be covered in this paper.

2.1.2 Why do ensemble methods work

Thomas G. Dietterich 2003 identified three significant challenges faced by single models that ensemble methods can help address: statistical, computational, and representational problems.

Firstly, the statistical problem occurs when there are too many possible models for the available training data. In this case, multiple models might fit the training data well, but the chosen model may not perform well on new, unseen data, resulting in overfitting. Ensemble methods help mitigate this issue by aggregating predictions from multiple models, thereby reducing the risk of overfitting and improving generalization (Thomas G. Dietterich 2003, Polikar 2006). According to Sagi and Rokach 2018, this advantage is especially pronounced when data is limited, which is the case in the dataset under analysis.

On the other hand, the computational problem arises when identifying the optimal solution becomes computationally infeasible, often requiring the use of approximate methods such as gradient descent. While these approaches can provide reasonably good solutions, they may not always achieve the true optimal result. Ensemble methods ameliorate this by combining the outputs of various heuristics (Thomas G. Dietterich 2003).

Lastly, the representational issue arises when the set of possible solutions does not include options that can closely match the true underlying pattern or function. Ensemble methods address this by broadening the range of possibilities by combining multiple solutions to achieve a more accurate approximation of the true function (Thomas G. Dietterich 2003, Polikar 2006). Sagi and Rokach 2018 included a useful visual example of the representational problem, as shown in figure A1 in the appendix A.1.

Thomas G. Dietterich 2003 emphasized that ensemble methods effectively reduce both bias, associated with the representational problem, and variance, which is related to overfitting and computational inefficiencies. This reduction enhances the overall performance and robustness of learning algorithms.

2.1.3 Main characteristics of a good ensemble

According to Surowiecki 2004, the wisdom of the crowd, an idea that large groups of people are collectively smarter than individual experts, is effective when criteria such as independence, decentralization, and diversity of opinions, are met. Applying these principles to ensemble methods in machine learning, however, requires nuanced consideration, as methods like sequential ensembles intentionally introduce interdependence among base-learners.

In this context, there are two criteria that primarily contribute to the success of a machine learning ensemble (Sagi and Rokach 2018, Perales-González et al. 2022). The foremost and perhaps most popular factor is its ability to leverage diversity, which involves heterogeneity among base-learners or data manipulation. Diversity is crucial in preventing excessive correlation among base-learners, as combining highly correlated base-learners offers no benefit over using a single model.

Secondly, the predictive performance of the base-learners should be as high as possible (Sagi and Rokach 2018, Perales-González et al. 2022). This principle originates from the work of Ali and Pazzani 1995, whose research demonstrated that there is a consistent, linear connection between reducing errors and achieving more uncorrelated error patterns among base-learners.

These two criteria are, however, often in conflict. When base-learners achieve low prediction errors, their learning hypotheses align more closely, leading to a reduction in diversity. Thus, there's a trade-off of higher accuracy often leading to less diversity. Balancing these factors is nevertheless crucial for the ensemble's performance (Rooney et al. 2004).

2.1.3.1 How to ensure diversity

The most popular way to introduce diversity in an ensemble is through data manipulation (Allende and Valle 2016), an approach often applied in homogeneous ensembles. In bagging, for instance, diversity is achieved through bootstrap aggregating, which generates different datasets by sampling with replacement from the original dataset (Breiman 1996), as mentioned in section 2.1.1 (Introduction to Ensemble Methods).

Unlike a homogeneous ensemble, the heterogeneous approach leverages the diversity among base-learners, thus benefiting from the interaction between different algorithms. Measuring diversity is, however, challenging due to the absence of a universally accepted formal definition (Kuncheva and Whitaker 2003). In their investigation, the authors examined ten different statistics for assessing diversity among binary classifier outputs, including the average pairwise correlation, all yielding similar outcomes.

The average pairwise correlation approach quantifies diversity by calculating the correlation coefficients between the predictions of every pair of base-learners in the ensemble (Oza et al. 2003). By averaging these coefficients across all possible pairs, a single value is obtained that reflects the overall diversity among the base-learners. A lower average correlation indicates greater diversity, suggesting that the base-learners are making different predictions on the same data points.

2.1.4 Aggregation Methods

Having established the importance of diversity and individual performance in creating robust ensemble models, the next crucial step is determining how these diverse predictions are synthesized into a single forecast. Since the key principle of ensemble learning is to improve overall performance by combining the predictions of multiple base-learners, the method of aggregation becomes essential as the final step in the process. The simple average is the simplest yet most commonly used approach for combining predictions (Zhou 2021), despite its sensitivity to outliers (Allende and Valle 2016).

This sensitivity to outliers led to a slight adjustment known as the trimmed mean, where the $\alpha\%$ of base-learners with the highest and lowest predictions are excluded from the computation. The work from Jose and Winkler 2008 showed that moderate trimming of 10–30%, i.e. removing 5–15% from each end, yields good results. The Winsorized mean, a variation of the trimmed mean that involves not removing but capping extreme values, with a Winsorizing of 15–45%, i.e. capping 7.5–22.5% on each side, is an aggregation method that also works well (Jose and Winkler 2008).

In addition to those aggregation methods mentioned, others include performance-based ones, such as linear combination and the outperformance approach (Bunn 1975) that will not be covered in this paper.

2.1.5 Selection of Key Ensemble Methods for This Study

Alongside the reasons for utilizing ensemble methods in time series forecasting outlined by Allende and Valle 2016, including reducing errors caused by faulty assumptions of an individual algorithm, the most compelling reason for the case at hand is their ability to reduce uncertainty across various factors. These encompass the choice of the optimal forecasting algorithm, the selection of the correct embed size, and the ability to accommodate diverse data patterns such as spiky and smoother sections. Considering both homogeneous and heterogeneous methods allows for leveraging their unique strengths. While homogeneous methods effectively address diverse data patterns such as spiky and smoother sections, heterogeneous methods build on this capability by further reducing uncertainty in the choice of the optimal forecasting algorithm through the complementary strengths of diverse base-learners. Consequently, this study will explore both homogeneous and heterogeneous ensemble approaches.

2.1.5.1 Bagging for Time Series Forecasting

Since its introduction, research has demonstrated the effectiveness of bagging in time series forecasting (Petropoulos, Rob J Hyndman, and Bergmeir 2018) establishing it as the key pillar for the homogeneous approach of this study. As outlined in section 2.1.1 (Introduction to Ensemble Methods), the spectrum of ensemble methods available is already very diverse, and, even within bagging itself, there is a wide range of approaches to choose from. Tüysüzoğlu and Birant 2020 listed 25 different bagging methods, ranging from Nice Bagging (Skurichina and Duin 1998) to Actively Balanced Bagging (Błaszczyszki and Stefanowski 2017), whilst introducing Enhanced Bagging themselves, increasing the pool of bagging methods even further. Despite this diversity, not all bagging methods are suitable for time series data, due to their inherent temporal dependencies. Breiman himself already emphasized that bagging is particularly useful for unstable procedures (Breiman 1996).

To extend bagging to time series data while preserving temporal dependencies, the Moving Block Bootstrap (MBB) was introduced by Künsch 1989. MBB begins with block construction, where the time series is divided into overlapping blocks of a fixed length l . The choice of l is crucial because it must be large enough to capture the dependencies within the time series but small enough to provide sufficient variability among the resampled blocks (Vogel and Shallcross 1996). The blocks are then randomly sampled with replacement to create new bootstrap samples (Künsch 1989). Lastly, a large number of bootstrap samples are generated by concatenating the randomly sampled blocks. By preserving the order and structure of the data within each block, MBB effectively maintains the temporal dependencies inherent in the original time series. Figure A2 in appendix A.2 visualizes a simple MBB example.

An important advancement in bagging for time series forecasting relevant to this study was achieved in 2015 by M. Oliveira and L. Torgo, who proposed a method to enhance predictor diversity by varying embed sizes among base-learners. Oliveira and Torgo 2015 demonstrated that this specific variant significantly outperformed standard bagging across fourteen real-world time series datasets. This work is particularly relevant to this research, as the embed sizes, as discussed in section 1.5.3.2.2 (Baseline Models), had remained an unresolved issue during the PBL project.

2.1.5.2 Dynamic Integration for Time Series Forecasting

Given the aforementioned uncertainty surrounding the optimal forecasting algorithm, highlighted by the presence of multiple well-suited algorithms on the PBL Pareto-frontier, illustrated in Figure 5, the analysis was expanded beyond the homogeneous approach of bagging to include heterogeneous approaches. Specifically, this study investigated and applied the concept of Dynamic Integration (DI) for regression, as proposed by Rooney et al. 2004, which encompasses Dynamic Selection (DS), Dynamic Weighting (DW), and Dynamic Weighting with Selection (DWS), to the task of time series forecasting.

DS focuses on choosing the best base-learner to make a prediction. As described in Rooney et al. 2004, base-learner selection can either be static, where the best-performing base-learner

according to the training error is selected to make a prediction, or dynamic, where base-learners are chosen to adapt to localized trends in the data by identifying instances in the training set that are similar to each test instance and selecting the base-learner that best suits these conditions. DS rests on the rationale that a model may outperform others within certain regions of the instance space, even if it is not the best on average over the entire dataset (Rooney et al. 2004).

In contrast to DS, DW does not rely on selecting a single base-learner but instead distributes influence across multiple base-learners with weights assigned based on each base-learner's recent performance, using the normalized inverse error as the weighting factor.

DWS extends DW by introducing a filtering mechanism, where base-learners that perform poorly on the most recent observations are excluded from contributing to the final prediction (Rooney et al. 2004). This approach further refines the ensemble by discarding poorly performing base-learners, thereby enhancing the robustness of the final output.

2.2 Methodology

2.2.1 Base-learner Choice

For both the homogeneous and heterogeneous approaches, the initial step before implementation involved choosing the base-learner, which refers to the algorithms and their specific configurations that form the foundation of the ensemble. The pool of base-learners comprised various configurations of six algorithms: Average, Weekday Average, ARIMA, SARIMA, Exponential Smoothing, and Theta. To ensure diversity within the ensemble, one specific configuration of each algorithm was chosen for each approach.

In bagging, there is no interaction or combination of different algorithms, as the diversity is created through the data manipulation approach of MBB. Thus, the choice of the base-learner configuration was driven purely by performance considerations. Given the project's specific priorities, with MAPE being the primary metric, configurations that optimize this metric were favored, while stability, captured by the SI, remained an important factor. This prioritization was reflected by declaring the normalized MAPE to be 1.5 times as important as the normalized SI, resulting in weightings of 60 and 40 percent, respectively - a weighting that was chosen in

collaboration with NOS. The resulting base-learner configurations given this weight allocation are displayed in the figure A3 in appendix A.3.1.

In the case of DI, the choice of base-learners was not solely based on performance but also on maximizing the diversity of predictions among them. In line with this, here, a novel strategy is proposed to increase the diversity between different base-learners, which builds upon the average pairwise correlation method introduced in section 2.1.3.1 (How to ensure diversity), but implemented in a dynamic fashion. It begins by ranking the algorithms according to their average performance across their respective Pareto-efficient configurations, as presented in table A4 (b) in appendix A.3.2. SARIMA, identified as the highest-ranked algorithm, was selected as the first base learner. The configuration chosen for the base learner optimally balanced the two evaluation metrics, maintaining 1.5 times higher importance of the normalized MAPE compared to the normalized SI. For the subsequent choices, the next algorithm in the ranking was considered, and the configuration that minimizes pairwise correlation with the already selected base-learner was chosen. Although the average pairwise correlation approach introduced in section 2.1.3.1 (How to ensure diversity) relies on predictions, an error-based approach yielded superior results and was therefore preferred. The chosen base-learners are depicted in figure A5 in appendix A.3.2.

2.2.2 Bagging

The core mechanism of the bagging approach revolves around the MBB, which creates multiple bootstrap datasets from the time series by sampling contiguous data blocks, as described in section 2.1.5.1 (Bagging for Time Series Forecasting). A rolling window was used to define the training dataset at each iteration. Specifically, the rolling window selected a fixed number of data points from the past, which served as the training set for making predictions for the next time point. This window slid forward, updating the training data dynamically as new observations became available. This allowed the model to remain adaptive to recent patterns and trends in the dataset.

The fixed number of data points from the past used in the rolling window was defined by two hyperparameters: the embed size of the model and a specified number of additional points.

Table A1 shows the hyperparameter space for the number of additional points, along with all other hyperparameters and their respective search space. A larger number of additional points enhances the diversity of the bootstrap samples, which is crucial for effective ensemble modeling as emphasized in section 2.1.3 (Main characteristics of a good ensemble). However, setting this number too high can introduce irrelevant historical data, diluting the importance of recent observations critical for accurate forecasting. This can result in models that overgeneralize and thus fail to capture details of the most current data patterns. Therefore, while the embed sizes were fixed based on the configuration from PBL, careful tuning of the additional points was vital for achieving optimal performance in the ensemble forecasting model. A visual representation of this rolling window approach given an embed size of three and three additional points for the test set can be seen in figure A6 in appendix A.4.

Once the bootstrap samples were created, each sample was processed through the selected base-learner to generate individual predictions for the next time step. Motivated by the work of Oliveira and Torgo 2015 the ensemble implemented can create diversity not only through the MBB approach but also by incorporating multiple embed sizes. The predictions were then aggregated to produce the ensemble prediction for the upcoming time step using either the mean, median, trimmed mean with a 10% trimming, or winsorized mean with 15% winsorization applied to both tails.

2.2.3 Dynamic Integration

Just like the bagging approach, the DI approach utilized a rolling window mechanism to define the training dataset at each iteration. Figure A7 in appendix A.4 depicts this mechanism for a model with an embed size of 5 and an evaluation window size of 3, on which the base-learners' performance is evaluated for respective selection or weighting. The evaluation set grew progressively until it reached its fixed size, which was later maintained by shifting one data point forward with each iteration. Unlike the bagging approach, no additional points were needed to enhance the training set since diversity was created through the heterogeneity of the base-learner.

In all three DI approaches, whether selecting the best base-learner (DS), assigning weights to

all base-learners (DW), or selecting specific base-learners and determining their weights (DWS) for the next prediction, the decisions were based solely on MAPE within the evaluation window, as it was the primary metric to be optimized. It is important to note that the evaluation approach for the interim decisions followed the overall evaluation approach described in section 1.5.3.3.5 (Evaluation Window) by excluding outliers from the evaluation window. This exclusion ensured that outliers did not affect the process of base-learner selection or weight assignment, allowing the decisions to be based entirely on the core data within the evaluation window.

2.2.4 Hyperparameter tuning

To optimize model performance in bagging and DI, a tailored grid search was applied, systematically testing hyperparameters specific to each ensemble method. As in the PBL program, outliers were excluded from metric calculations to maintain the integrity of the MAPE results. MAPE was, however, not the only evaluation metric utilized. SI was also applied, as in the PBL project, with both metrics set as objectives for minimization.

For bagging, the grid search evaluated a range of embed sizes, block sizes, bootstrap sample counts, and aggregation methods, with additional options to expand the training set. For DI, the grid search focused on embed sizes and the number of base-learners making predictions. The latter parameter applied exclusively to DWS where only selected base-learners contributed to the forecast. This is because in DS, the prediction was made by a single base-learner, whereas in DW, all base-learners contributed. Specific values for each parameter are detailed in Table A1 in appendix A.5.

2.3 Results & Discussion

In alignment with the previous project, the outcomes of this study were visualized using the overall Pareto frontier approach and compared to the baseline, which encompassed the entire Pareto frontier from the PBL project. Emphasis was placed on the deployable solution, the SARIMA [(2, 0, 1), (0, 0, 1, 20)] model with an embed size of 20, which served as a key reference point within this frontier.

2.3.1 Bagging Results

Figures A8 to A10 in appendix A.6.1 present the results of different bagging strategies, with the base-learner choice weighted by a factor of 1.5 in favor of MAPE relative to SI. While figure A8 depicts bagging algorithms that utilized only bootstrapped samples generated with a single embed size, figure A9 contrasts this approach by showing bagging algorithms that used bootstrapped samples generated with a combination of multiple embed sizes only. Figure A10 builds further on these methodologies by representing bagging algorithms that incorporated both approaches, thereby synthesizing the two strategies.

The results consistently show that each bagging approach, regardless of the base-learner or embed size variant used, closely approached or exceeded the PBL Pareto frontier. This proximity to the benchmark highlights that these models consistently achieved performance levels comparable to the best results from the previous project, with minimal compromise in either MAPE or the SI. Notably, each individual Pareto frontier derived from a bagging approach, regardless of the base-learner or embed-size strategy employed, surpassed the PBL frontier in the region characterized by relatively higher MAPE and lower SI values. Thus, the bagging approach proved effective, particularly in scenarios where balance skewed more towards stability with slightly higher error rates.

Furthermore, even in the region characterized by relatively higher SI and lower MAPE values, the bagging approaches with Weekday Average and Exponential Smoothing as base-learners outperformed the PBL frontier regardless of the embed size applied. The deployable solution from the previous project was the only one to remain clearly unmatched, marking it as a critical point even on the newly established Pareto frontier.

This new frontier comprised configurations from the two aforementioned algorithms, the Weekday Average and Exponential Smoothing, along with the deployable solution from the previous project. Additionally, the point representing the lowest MAPE from the PBL project was included, although it was not significant in shaping the frontier due to the presence of a nearby point with a very similar MAPE but a substantially lower SI value in all three cases.

While utilizing multiple embed sizes appeared to be beneficial when compared to the PBL Pareto frontier, it did not outperform the single embed size approach. Using a single embed size proved to be the most effective approach, as demonstrated by the frequent appearance of embed sizes 10 and 15 in the Pareto-efficient outcomes across all embed-size strategies, shown in figure A10. Out of 116 Pareto-efficient points in this figure, the embed sizes 10 and 15 appeared over 55% and 40% of times, respectively, while the embed size 20 occurred only once. The combination [10, 15] stood as the sole representation of an embed-size combination, yet it appeared only three times.

2.3.2 Dynamic Integration Results

Figure A11 in appendix A.6.2 shows the Pareto frontiers of each approach within the landscape of DI. The frontier resulting from dynamically selecting the best base-learner, DS, outperformed the PBL frontier in the region characterized by higher MAPE and lower SI, similar to the bagging approaches. However, the two approaches that assign weights to base-learners, DW and DWS, were the only ones to surpass the deployable solution from the PBL project. While the DW approach achieved only a marginal improvement in both evaluation metrics, the DWS approach demonstrated significant enhancements in both MAPE and SI, resulting in a substantial shift in the Pareto frontier. Thus, by exclusively utilizing models from the DI approach, the entire PBL frontier was surpassed. The resulting Pareto-frontier is predominantly shaped by the DWS approach, with the DS approach contributing a single point in the higher MAPE and lower SI region.

Overall, the DI approach demonstrated its capacity to achieve superior performance in balancing accuracy and stability, with the DWS method outperforming other strategies in most cases and forming the foundation of the new Pareto frontier.

2.3.3 Final Model Decision and Performance Evaluation

Figure 7a presents the individual Pareto frontiers for both the bagging and DI methods. In the new Pareto frontier represented by the red dotted line in figure 7b, no points from the PBL frontier are included. Instead, it consists entirely of points from DWS and bagging. Notably, only

the Exponential Smoothing and Weekday Average configurations from the bagging approach contributed to the overall Pareto frontier, while other configurations did not meet the performance criteria required for inclusion. When applied directly to the initial time series dataset without bagging, however, both Weekday Average and Exponential Smoothing ranked among the worst-performing algorithms, as illustrated in figure A4 in appendix A.3.2. This outcome could be explained by the fact that the Weekday Average and Exponential Smoothing models disproportionately suffered during transitional periods in performance when applied individually to the time series dataset. This might be mitigated by bagging, which helps stabilize their performance by training an ensemble of base-learners on bootstrapped subsets of the data and aggregating their predictions (Breiman 1996). In contrast, the effect of bagging given base-learners like ARIMA and SARIMA is not as pronounced, as those inherently exhibit lower variance in their predictions within this context.

Building on the previously mentioned weighting scheme for metric prioritization, the final model was selected by weighting the normalized MAPE as 1.5 times more important than the normalized SI, resulting in the red-outlined point. This point belongs to a configuration of the DWS approach, in which three base-learners were selected in each iteration and weighted according to their performance on the last nine predictions, resulting in a MAPE of 6.21 and an SI of 1.15. This configuration demonstrated a substantial improvement, achieving nearly a 10% reduction in MAPE and an over 20% reduction in SI compared to the deployable solu-

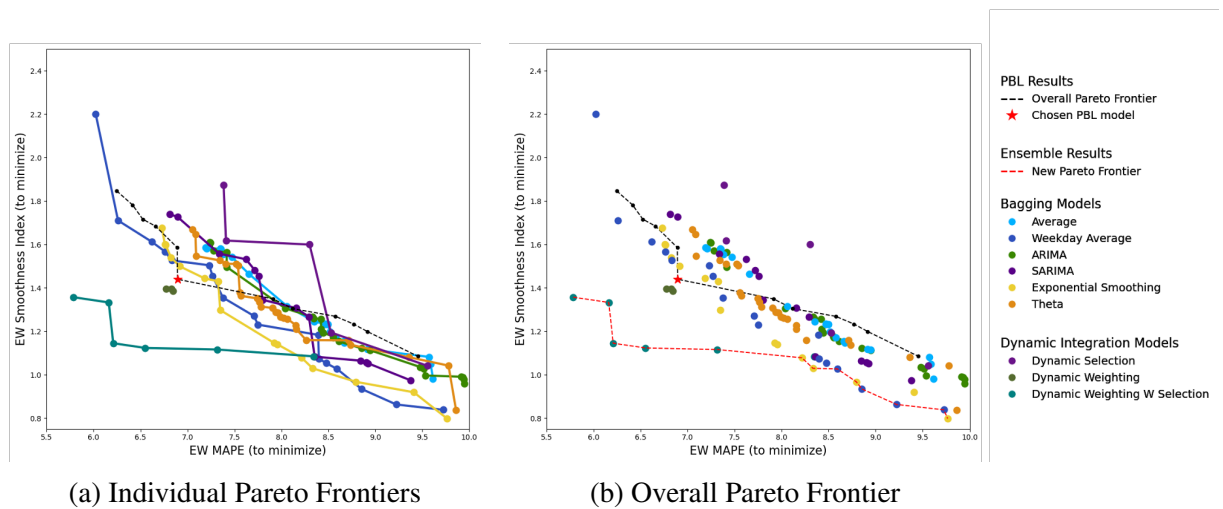


Figure 7: Pareto frontier using bagging and DI models

tion from the PBL project, which had a MAPE of 6.89 and an SI of 1.44. Notably, each of the six base-learner options contributed meaningfully to the predictive outcomes, as shown in figure A12. Among these, the DWS most frequently selected the Weekday Average and Exponential Smoothing methods to contribute to predictions across iterations, which notably aligns with their strong performance in the bagging approach. Building on the previous hypothesis, this suggests that although Weekday Average and Exponential Smoothing performed poorly during transitions, they excelled before and after these periods, outperforming other algorithms like ARIMA, SARIMA, and Theta. This characteristic is confirmed by figure A13 in appendix A.6.3, which illustrates the contribution of each base-learner algorithm over time, showing the weights allocated by the DWS algorithm for each prediction in panel (a) and presenting the ranked contributions in panel (b), whereby the highest contributing base-learner was assigned rank 1. As a result, these models became particularly effective when dynamically integrated, as their strengths could be harnessed effectively outside of transitional periods, enhancing overall ensemble performance.

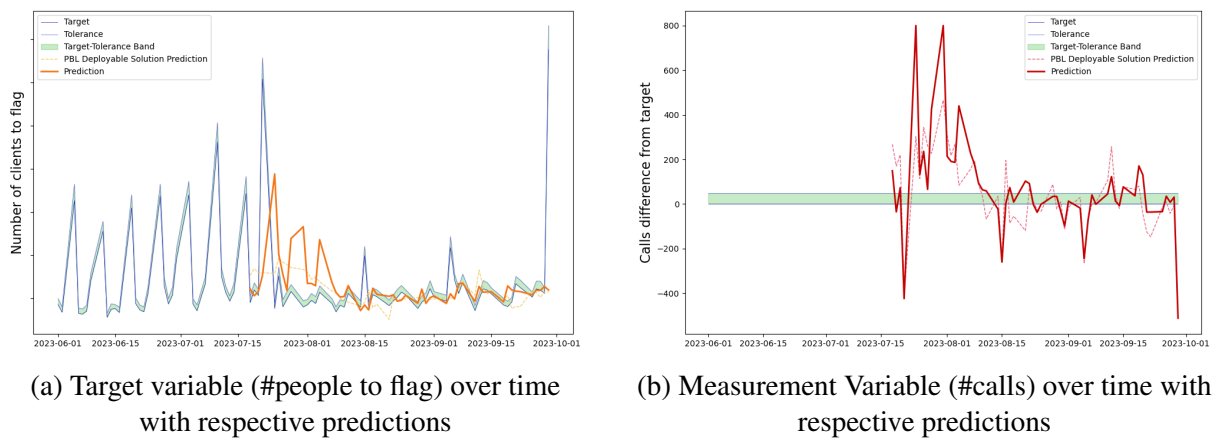


Figure 8: Chosen DWS Model Performance

Figures 8a and 8b illustrate the performance of the model on the target variable and measurement variable, respectively. In contrast to the deployable model from PBL, the DWS model possessed the capability of capturing rapid fluctuations in values, allowing it to respond to abrupt changes from one period to the next. An initial examination of figure 8a suggests that the PBL model adjusted seamlessly to the smoother sections of the time series, whereas this transition appeared to have a more pronounced negative impact on the performance of the DWS model. However, upon closer investigation, it appears that whilst the DWS model quickly adapted to

the smoother section of the time series, the deployable PBL model continued to decrease its predictions. Further, once the DWS model adapted to the smooth section, it also hit the target-tolerance band more often and generally made more accurate predictions, as reflected in the lower MAPE value and visualized in figure 8b.

2.4 Limitations & Future Work

This work, like the previous PBL project, was significantly limited by the short time series available for modeling. While exploring ensembles utilizing multivariate base-learners could provide additional insights, the limited dataset and their underperformance compared to univariate models during the PBL project prompted a focus on univariate base-learners for the ensemble approaches. Future work could address this limitation by incorporating larger datasets and further investigating the potential of multivariate base-learners in ensemble approaches. Moreover, larger datasets would enable the exploration of additional aggregation methods in the bagging approach, such as linear combination and the outperformance approach. These methods were not feasible in this study due to the nature of the rolling window coupled with MBB and the limited data size.

Next to the types of base-learners and aggregation methods, this limitation also made it infeasible to implement a fully dynamic approach regarding the base-learner selection for DS or DWS, introduced in section 2.1.5.2 (Dynamic Integration for Time Series Forecasting). By integrating this, the ensemble methods DS and DWS could potentially capture the complexities in the data more effectively, motivating further exploration of dynamic base-learner selection in future research.

Furthermore, the dynamics of base-learner contributions over time within the DWS algorithm, as discussed in section 2.3.3 (Final Model Decision and Performance Evaluation) and shown in Figure A13, warrant further investigation to refine the understanding of how specific base-learners excel in distinct phases of the time series. This could include exploring alternative ensemble strategies that dynamically adapt to transitions, such as incorporating specialized transition-focused base-learners, or modifying weighting schemes to account for transition-

induced volatility.

Lastly, future work could investigate the transferability of the discussed ensemble methods and the novel dynamic average pairwise correlation approach for the choice of base-learners within the DI framework to other domains with diverse time series characteristics. This application could provide providing deeper insights into their robustness and adaptability beyond call-center data.

2.5 Conclusion

This study highlighted the effectiveness of ensemble learning methods for time series forecasting in the context of small datasets, exploring both a homogeneous approach (bagging), and a heterogeneous approach (DI). Bagging demonstrated strong performance, generally producing models with enhanced stability (SI) with a modest trade-off in accuracy (MAPE), particularly when utilizing base-learners based on Exponential Smoothing and Weekday Average. Conversely, DI, especially the DWS approach, excelled when accuracy was prioritized over stability, offering a more precision-driven solution. The integration of performance-based weighting and selective inclusion of base-learners evidently further enhanced DI's ability to address the challenges of forecasting with limited data. The chosen DWS configuration reduced MAPE by 10% and SI by over 20% compared to the deployable solution from the PBL project, substantially outperforming traditional single-model approaches.

A key novelty of this study is the introduction of dynamic error-based pairwise correlation to ensure diversity among base-learners in DI, minimizing error correlations and enhancing robustness. This approach differs from the average pairwise correlation approach mainly used for classification tasks by dynamically selecting configurations that not only minimize correlation with already chosen base-learners, but also prioritize strong performance, proving effective in this application. Furthermore, while DI has been applied in regression tasks, its application to time series forecasting, particularly in scenarios with small datasets, has not been extensively researched. This study extended its application to this new context, demonstrating its potential for further exploration in this area.

In conclusion, both bagging and DI demonstrated their effectiveness as ensemble learning strategies for time series forecasting, each offering distinct advantages tailored to the specific characteristics of the dataset in terms of the stability-accuracy trade-off. These findings highlight the power of ensemble learning to address the challenges of small-dataset time series forecasting, balancing accuracy and stability while adapting to diverse forecasting scenarios. However, the DI approach, specifically the DWS configuration, which dynamically selected the top three base-learners from the six available and assigned their weights based on performance over the last nine predictions, achieved the best balance between the two metrics, significantly outperforming the PBL approach in both. Based on this outcome, it is recommended that NOS adopt the DWS approach to enhance the accuracy and reliability of their forecasts, improving the prioritization of clients likely to make calls for expedited resolution by the Detractor Squad.

References

- Ali, Kamal M and Michael J Pazzani (Sept. 1995). *On the link between error correlation and error reduction in decision tree ensembles*. Tech. rep. UC Irvine: Donald Bren School of Information and Computer Sciences. URL: <https://escholarship.org/uc/item/9k55v622> (visited on 12/09/2024).
- Allende, Héctor and Carlos Valle (Oct. 2016). “Ensemble Methods for Time Series Forecasting”. In: *Claudio Moraga: A Passion for Multi-Valued Logic and Soft Computing*. Ed. by Rudolf Seising and Héctor Allende-Cid. Springer International Publishing, pp. 217–232. ISBN: 978-3-319-48317-7. DOI: 10.1007/978-3-319-48317-7_13.
- Assimakopoulos, Vassilis and Konstantinos Nikolopoulos (Dec. 2000). “The theta model: a decomposition approach to forecasting”. In: *International Journal of Forecasting* 16.4, pp. 521–530. DOI: 10.1016/S0169-2070(00)00066-2.
- Bastianin, Andrea, Marzio Galeotti, and Matteo Manera (June 2019). “Statistical and economic evaluation of time series models for forecasting arrivals at call centers”. In: *Empirical Economics* 57.3, pp. 923–955. DOI: 10.1007/s00181-018-1475-y.
- Bates, John M and Clive WJ Granger (Dec. 2017). “The Combination of Forecasts”. In: *Journal of the Operational Research Society* 20.4. Originally published in September, 1969, pp. 451–468. DOI: 10.1057/jors.1969.103.
- Bergmeir, Christoph and José M. Benítez (2012). “On the Use of Cross-Validation for Time Series Predictor Evaluation”. In: *Information Sciences* 191, pp. 192–213. DOI: 10.1016/j.ins.2011.12.028.
- Berrar, Daniel et al. (2019). “Cross-validation”. In: *Encyclopedia of Bioinformatics and Computational Biology* 1, pp. 542–545.

- Błaszczczyński, Jerzy and Jerzy Stefanowski (June 2017). “Actively Balanced Bagging for Imbalanced Data”. In: *Foundations of Intelligent Systems*. Ed. by Marzena Kryszkiewicz et al. Springer International Publishing, pp. 271–281. DOI: 10.1007/978-3-319-60438-1_27.
- Box, George EP et al. (Mar. 2016). “Time series analysis: forecasting and control”. In: *Journal of Time Series Analysis*. Ed. by Granville Tunnicliffe Wilson. Vol. 37. 5. John Wiley & Sons, pp. 712. DOI: 10.1111/jtsa.12194.
- Breiman, Leo (Aug. 1996). “Bagging predictors”. In: *Machine learning* 24. Ed. by Ross Quinlan, pp. 123–140. DOI: 10.1007/BF00058655.
- Brown, R.G. (Jan. 1959). *Statistical Forecasting for Inventory Control*. Vol. 2. New York: McGraw-Hill, pp. 443–473. URL: <https://books.google.es/books?id=oKI8AAAAIAAJ>.
- Bühlmann, Peter (2012). “Bagging, Boosting, and Ensemble Methods”. In: *Handbook of Computational Statistics: Concepts and Methods*, pp. 985–1022. DOI: 10.1007/978-3-642-21551-3_31.
- Bunn, Derek W (Dec. 1975). “A Bayesian Approach to the Linear Combination of Forecasts”. In: *Journal of the Operational Research Society* 26.2, pp. 325–329. DOI: 10.1057/jors.1975.67.
- Cerqueira, Vitor, Luis Torgo, and Igor Mozetič (2020). “Evaluating Time Series Forecasting Models: An Empirical Study on Performance Estimation Methods”. In: *Machine Learning* 109.11, pp. 1997–2028. DOI: 10.1007/s10994-020-05910-7.
- Chakraborty, Kanad et al. (Nov. 1992). “Forecasting the behavior of multivariate time series using neural networks”. In: *Neural Networks* 5.6, pp. 961–970. DOI: 10.1016/s0893-6080(05)80092-9.

- Chatfield, Chris (Oct. 2000). *Time-series forecasting*. 1. Boca Raton: Chapman and Hall/CRC. DOI: 10.1201/9781420036206.
- Chen, Tianqi and Carlos Guestrin (Mar. 2016). “Xgboost: A scalable tree boosting system”. In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.
- Connor, J.T., R.D. Martin, and L.E. Atlas (Mar. 1994). “Recurrent neural networks and robust time series prediction”. In: *IEEE Transactions on Neural Networks* 5.2, pp. 240–254. DOI: 10.1109/72.279188.
- Davey, AM and BE Flores (Sept. 1993). “Identification of seasonality in time series: A note”. In: *Mathematical and Computer Modelling* 18.6, pp. 73–81. DOI: 10.1016/0895-7177(93)90126-J.
- De Gooijer, Jan G. and Rob J. Hyndman (Jan. 2006). “25 years of time series forecasting”. In: *International Journal of Forecasting* 22.3, pp. 443–473. DOI: 10.1016/j.ijforecast.2006.01.001.
- Dietterich, Thomas G (Jan. 2000). “Ensemble Methods in Machine Learning”. In: *Multiple Classifier Systems*. Springer Berlin Heidelberg, pp. 1–15. DOI: 10.1007/3-540-45014-9_1.
- (2003). “Ensemble learning”. In: *The Handbook of Brain Theory and Neural Networks*. Ed. by Michael A. Arbib. Vol. 2. Originally published in September, 2002. MIT Press, pp. 405–408. ISBN: 9780262011976.
- Effron, Bradley and Robert J. Tibshirani (May 1994). *An Introduction to the Bootstrap*. Chapman and Hall/CRC. DOI: 10.1201/9780429246593.

- Ehrgott, Matthias (2012). “Vilfredo Pareto and multi-objective optimization”. In: *Doc. math* 8, pp. 447–453.
- Falessi, Davide et al. (2020). “On the Need of Preserving Order of Data When Validating Within-Project Defect Classifiers”. In: *Empirical Software Engineering* 25, pp. 4805–4830. DOI: 10.1007/s10664-020-09858-3.
- Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (Nov. 2016). *Deep Learning*. Cambridge: MIT Press, p. 775. DOI: 10.5555/3086952.
- Hamilton, James D. (Dec. 1994). *Time series analysis*. Princeton: De Gruyter and Jstor. DOI: 10.1515/9780691218632.
- Healy, M.J.R. and R.G. Brown (Jan. 1964). “Smoothing, Forecasting and Prediction of Discrete Time Series.” In: *Journal of the Royal Statistical Society Series A (General)* 127.2, p. 292. DOI: 10.2307/2344012.
- Hjorth, J. S. Urban (1994). *Computer intensive statistical methods. Validation, model selection, and bootstrap*. London: Chapman and Hall/CRC. DOI: 10.1201/9781315140056.
- Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). “Long Short-term Memory”. In: *Neural Computation* 9.8, pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.
- Holt, Charles C. (Jan. 2004). “Forecasting seasonals and trends by exponentially weighted moving averages”. In: *International Journal of Forecasting* 20.1, pp. 5–10. DOI: 10.1016/j.ijforecast.2003.09.015.
- Hyndman, Rob and George Athanasopoulos (July 2012). *Principles of Business Forecasting by Keith Ord & Robert Fildes Forecasting: Principles and Practice*.
- Hyndman, Rob J. and George Athanasopoulos (2018). *Forecasting: Principles and Practice*. 2nd. Melbourne: OTexts. DOI: 10.5281/zenodo.3740164.

- Jose, Victor Richmond R and Robert L Winkler (Mar. 2008). “Simple robust averages of forecasts: Some empirical results”. In: *International journal of forecasting* 24.1, pp. 163–169. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2007.06.001.
- Koopman, Siem Jan and Kai Ming Lee (Mar. 2009). “Seasonality with Trend and Cycle Interactions in Unobserved Components Models”. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 58.4, pp. 427–448. DOI: 10.1111/j.1467-9876.2009.00661.x.
- Korstanje, Joos (July 2021a). *Advanced Forecasting with Python*. New York: Apress.
- (July 2021b). *Advanced forecasting with Python. With State-of-the-Art-Models Including LSTMs, Facebook’s Prophet, and Amazon’s DeepAR*. 1. Berkeley: Apress Berkeley, CA. DOI: 10.1007/978-1-4842-7150-6.
- Kuncheva, Ludmila I. and Christopher J. Whitaker (May 2003). “Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy”. In: *Machine learning* 51, pp. 181–207. DOI: 10.1023/A:1022859003006.
- Künsch, Hans R (Sept. 1989). “The Jackknife and the Bootstrap for General Stationary Observations”. In: *The Annals of Statistics* 17.3, pp. 1217–1241. ISSN: 00905364, 21688966. URL: <http://www.jstor.org/stable/2241719> (visited on 12/09/2024).
- LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (2015). “Deep Learning”. In: *Nature* 521.7553, pp. 436–444. DOI: 10.1038/nature14539.
- Lim, Bryan and Stefan Zohren (Feb. 2021). “Time-series forecasting with deep learning: a survey”. In: *Philosophical Transactions of the Royal Society A* 379.2194. DOI: 10.1098/rsta.2020.0209.
- Oliveira, Mariana and Luis Torgo (Nov. 2015). “Ensembles for Time Series Forecasting”. In: *Proceedings of the Sixth Asian Conference on Machine Learning*. Ed. by Dinh Phung and

- Hang Li. Vol. 39. Proceedings of Machine Learning Research. PMLR, pp. 360–370. URL: <https://proceedings.mlr.press/v39/oliveira14.html> (visited on 12/09/2024).
- Opitz, David and Richard Maclin (Aug. 1999). “Popular Ensemble Methods: An Empirical Study”. In: *Journal of Artificial Intelligence Research* 11, pp. 169–198. DOI: 10.1613/jair.614.
- Oza, Nikunj C. et al. (June 2003). “Classification of Aircraft Maneuvers for Fault Detection”. In: *Multiple Classifier Systems*. Springer Berlin Heidelberg, pp. 375–384. DOI: 10.1007/3-540-44938-8_38.
- Perales-González, Carlos et al. (Aug. 2022). “Global Negative Correlation Learning: A Unified Framework for Global Optimization of Ensemble Models”. In: *IEEE Transactions on Neural Networks and Learning Systems* 33.8, pp. 4031–4042. DOI: 10.1109/TNNLS.2021.3055734.
- Petropoulos, Fotios, Daniele Apiletti, et al. (Sept. 2022). “Forecasting: theory and practice”. In: *International Journal of Forecasting* 38.3, pp. 705–871. DOI: 10.1016/j.ijforecast.2021.11.001.
- Petropoulos, Fotios, Rob J Hyndman, and Christoph Bergmeir (July 2018). “Exploring the sources of uncertainty: Why does bagging for time series forecasting work?” In: *European Journal of Operational Research* 268.2, pp. 545–554. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2018.01.045.
- Pindyck, Robert S et al. (Feb. 2018). *Microeconomics*. 9. Upper Saddle River: Pearson. ISBN: 978-0134184241.
- Płaza, Mirosław and Łukasz Pawlik (Mar. 2021). “Influence of the contact center systems development on key performance indicators”. In: *IEEE Access* 9, pp. 44580–44591. DOI: 10.1109/ACCESS.2021.3066801.

- Polikar, Robi (Sept. 2006). “Ensemble based systems in decision making”. In: *IEEE Circuits and Systems Magazine* 6.3, pp. 21–45. DOI: 10.1109/MCAS.2006.1688199.
- Purba, Johana Sihol Marito and Juliza Hidayati (2020). “Performance improvement strategies to increase call center service level: a literature review”. In: *IOP Conference Series: Materials Science and Engineering* 801.1, p. 012147. DOI: 10.1088/1757-899X/801/1/012147.
- Rooney, Niall et al. (2004). “Dynamic Integration of Regression Models”. In: *Multiple Classifier Systems*. Ed. by Fabio Roli, Josef Kittler, and Terry Windeatt. Springer Berlin Heidelberg, pp. 164–173. DOI: 10.1007/978-3-540-25966-4_16.
- Sagi, Omer and Lior Rokach (Feb. 2018). “Ensemble learning: A survey”. In: *WIREs Data Mining and Knowledge Discovery* 8.4, e1249. DOI: 10.1002/widm.1249.
- Sakib, Mohd, Suhel Mustajab, and Mahfooz Alam (Nov. 2024). “Ensemble deep learning techniques for time series analysis. A Comprehensive Review, Applications, Open Issues, Challenges, and Future Directions”. In: *Cluster Computing*. Vol. 28. 1. Springer. DOI: 10.1007/s10586-024-04684-0.
- Schapire, Robert E (June 1990). “The strength of weak learnability”. In: *Machine learning* 5, pp. 197–227. DOI: 10.1007/BF00116037.
- Shmueli, Galit and Julia Polak (Feb. 2024). *Practical Time Series Forecasting with R: A Hands-On Guide*. Third. New York: Axelrod Schnall Publishers, p. 252. ISBN: 9780997847949. URL: <https://www.barnesandnoble.com/w/practical-time-series-forecasting-with-r-julia-polak/1144996927> (visited on 12/09/2024).
- Shumway, Robert H. and David S. Stoffer (2000). *Time Series Analysis and Its Applications*. Vol. 3. New York: Springer. DOI: 10.1007/978-1-4757-3264-5.

- Shumway, Robert H. and David S. Stoffer (2017). “ARIMA Models”. In: *Time Series Analysis and Its Applications: With R Examples*, pp. 75–163. DOI: 10.1007/978-3-319-52452-8.
- Skurichina, Marina and Robert PW Duin (July 1998). “Bagging for linear classifiers”. In: *Pattern Recognition* 31.7, pp. 909–930. ISSN: 0031-3203. DOI: 10.1016/S0031-3203(97)00110-6.
- Surowiecki, J (June 2004). *The wisdom of crowds. why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. Doubleday. ISBN: 0-385-50386-5.
- Taylor, James W. (Dec. 2007). “A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center”. In: *Management Science* 54.2, pp. 253–265. DOI: 10.1287/mnsc.1070.0786.
- Taylor, Sean J and Benjamin Letham (Apr. 2018). “Forecasting at Scale”. In: *The American Statistician* 72.1, pp. 37–45. DOI: 10.1080/00031305.2017.1380080.
- Torres, José F. et al. (Feb. 2021). “Deep Learning for Time Series Forecasting: A Survey”. In: *Big Data* 9.1, pp. 3–21. DOI: 10.1089/big.2020.0159.
- Tukey, John W. (Jan. 1977). *Exploratory data analysis*. Reading: Pearson. ISBN: 978-0201076165.
- Tüysüzoğlu, Göksu and Derya Birant (July 2020). “Enhanced bagging (eBagging): A novel approach for ensemble learning”. In: *International Arab Journal of Information Technology* 17.4, pp. 515–528. DOI: 10.34028/iajit/17/4/10.
- Vagropoulos, Stylianos I. et al. (2016). “Comparison of SARIMAX, SARIMA, Modified SARIMA and ANN-Based Models for Short-Term PV Generation Forecasting”. In: *2016 IEEE International Energy Conference (ENERGYCON)*. IEEE, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7513987.

- Vishwas, BV and Ashish Patel (Aug. 2020). *Hands-on Time Series Analysis with Python. From Basics to Bleeding Edge Techniques*. 1. Berkeley: Apress Berkeley, CA. DOI: 10.1007/978-1-4842-5992-4.
- Vogel, Richard M. and Amy L. Shallcross (June 1996). “The moving blocks bootstrap versus parametric time series models”. In: *Water Resources Research* 32.6, pp. 1875–1882. DOI: 10.1029/96WR00928.
- Wang, Feng and Joey Aviles (May 2023). “Contrasting Univariate and Multivariate Time Series Forecasting Methods for Sales: A Comparative Analysis”. In: *Applied Science and Innovative Research* 7.2, p127. DOI: 10.22158/asir.v7n2p127.
- Wang, Yan and Yuankai Guo (Mar. 2020). “Forecasting method of stock market volatility in time series data based on mixed model of ARIMA and XGBoost”. In: *China Communications* 17.3, pp. 205–221.
- Wang, Yichang (Sept. 2021). “Interpretable time series classification”. PhD thesis. Université Rennes 1. URL: <https://theses.hal.science/tel-03509607> (visited on 12/09/2024).
- Winkler, Robert L and Spyros Makridakis (Mar. 1983). “The Combination of Forecasts”. In: *Journal of the Royal Statistical Society: Series A (General)* 146.2, pp. 150–157. ISSN: 00359238, 23972327. URL: <http://www.jstor.org/stable/2982011> (visited on 12/09/2024).
- Winters, Peter R. (Apr. 1960). “Forecasting sales by exponentially weighted moving averages”. In: *Management science* 6.3, pp. 324–342. DOI: 10.1287/mnsc.6.3.324.
- Wu, Wei Biao and Zhibiao Zhao (May 2007). “Inference of trends in time series”. In: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 69.3, pp. 391–410. DOI: 10.1111/j.1467-9868.2007.00594.x.

- Yao, Hui et al. (Apr. 2023). “Advanced industrial informatics towards smart, safe and sustainable roads: A state of the art”. In: *Journal of traffic and transportation engineering (English edition)* 10.2, pp. 143–158. DOI: 10.1016/j.jtte.2023.02.001.
- Yildiz, Cem Yarkin and O Erhun Kundakcioglu (Feb. 2024). “Optimization for Time Series Decomposition”. In: *Encyclopedia of Optimization*. Springer Nature, pp. 1–9. DOI: 10.1007/978-3-030-54621-2_881-1.
- Yule, George Udny (Jan. 1927). “VII. On a method of investigating periodicities disturbed series, with special reference to Wolfer’s sunspot numbers”. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 226.636-646, pp. 267–298. DOI: 10.1098/rsta.1927.0007.
- Zhang, Cha and Yunqian Ma (Feb. 2012). *Ensemble Machine Learning. Methods and Applications*. Vol. 144. Springer New York, NY. DOI: 10.1007/978-1-4419-9326-7.
- Zhang, Kaimeng, Chi Tim Ng, and Myung Hwan Na (Apr. 2020). “Real time prediction of irregular periodic time series data”. In: *Journal of Forecasting* 39.3, pp. 501–511. DOI: 10.1002/for.2637.
- Zhou, Zhi-Hua (May 2012). *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC press. DOI: 10.1201/b12207.
- (Aug. 2021). “Ensemble learning”. In: *Machine Learning*. Springer Singapore, pp. 181–210. ISBN: 978-981-15-1967-3. DOI: 10.1007/978-981-15-1967-3_8.

A Appendix

A.1 Visualization of the Representation Problem

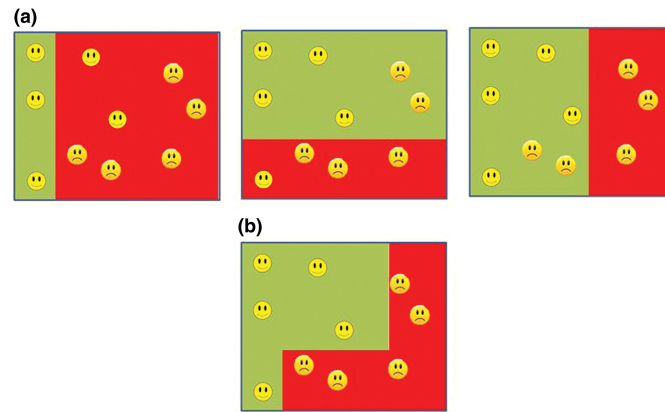


Figure A1: Visualization of the representation problem (Sagi and Rokach 2018)

Figure A1 demonstrates an example of the representational problem, adapted from Sagi and Rokach 2018. In this example, a binary classification task is illustrated using two attributes: a happy face and a sad face. Three different decision stumps were used, each of which misclassifies some data points, as shown in part (a). However, when these decision stumps are combined, they achieve perfect classification, as depicted in (b).

A.2 Visualization of the MBB approach

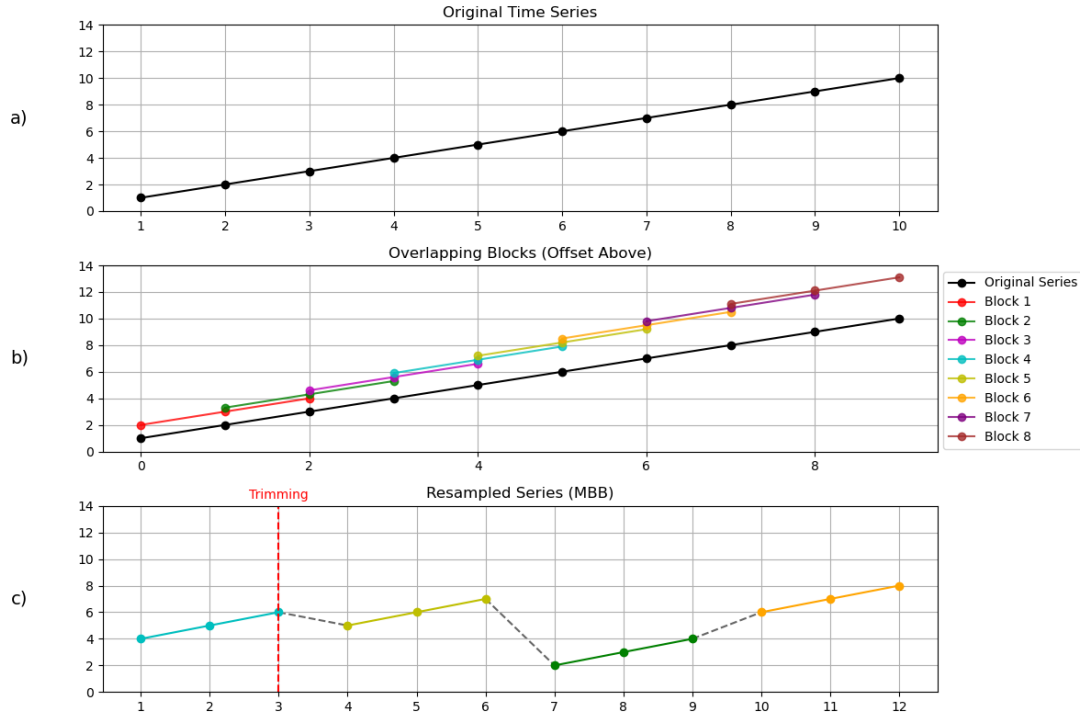


Figure A2: Example of MBB with a blocksize of 3

Figure A2 provides a step-by-step visualization of the MBB process, using a simple time series and a block size of three to illustrate the method. In part a) of the figure, the original time series is displayed with labeled data points to emphasize their sequential order and temporal structure. Below this, in part b), overlapping blocks of three data points are constructed by sliding forward one observation at a time, ensuring that each block retains the temporal dependency present within the original series. The blocks are depicted above the original series.

Part c) of the figure then demonstrates the process of sampling these blocks randomly with replacement. Each sampled block is shown with visual distinctions and labeled to indicate its position within the original time series, highlighting the randomness introduced in the selection. Some blocks may appear multiple times or be skipped entirely, providing variability in the resampling process. The sampled blocks are then concatenated to form a new bootstrap replicate of the time series. Caused by the concatenation, the length of the MBB series is longer than the original series, implying the need for trimming. Following trimming the randomly resampled series has the same length as the original series.

A.3 Choice for Base-Learners

A.3.1 Base-Learners Bagging

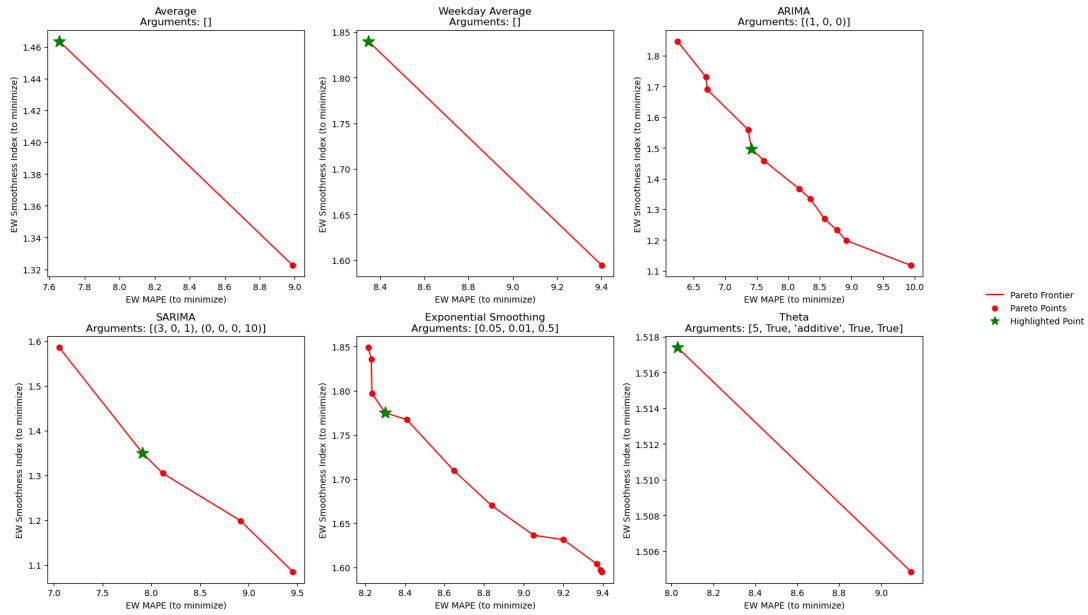
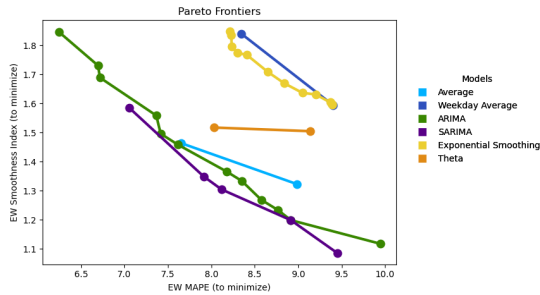


Figure A3: Individual Pareto Frontiers for Bagging given a Factor of 1.5

A.3.2 Base-Learners Dynamic Integration



(a) Individual Pareto frontiers from the PBL models without capping

Algorithm	Rank
SARIMA	1
ARIMA	2
Average	3
Theta	4
Exponential Smoothing	5
Weekday Average	6

(b) Ranks of univariate algorithms in PBL

Figure A4: Ranking algorithms by average performance across Pareto-efficient configurations

The ranking is determined by averaging a weighted performance score for each algorithm, where the normalized Mean Absolute Percentage Error (MAPE) is assigned a weight 1.5 times greater than the normalized Stability Index (SI). Algorithms are subsequently ranked based on their respective weighted average scores.

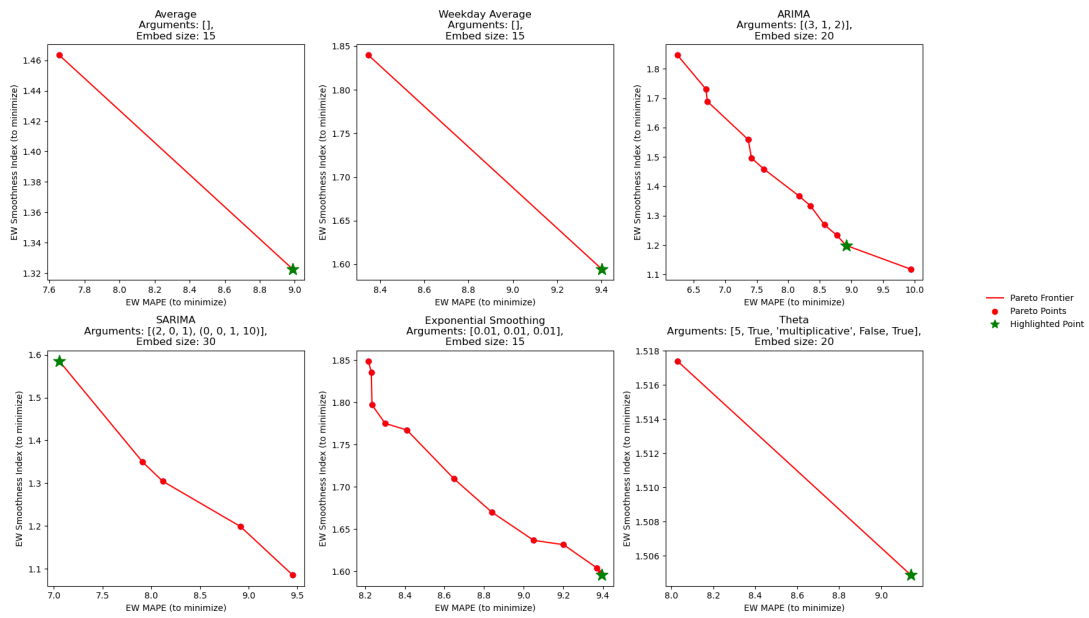


Figure A5: Individual Pareto Frontiers for DI given the dynamic, error-based approach

A.4 Visualization of the rolling window approach

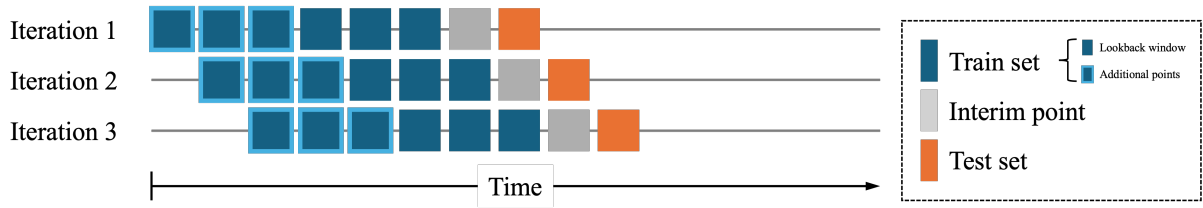


Figure A6: Example of the rolling window mechanism for Bagging

Figure A6 illustrates the rolling window approach for the train-test split used in bagging. Each row represents a new iteration, where the dark blue blocks indicate the training data. The dark blue blocks with light blue borders represent additional points, the number of which is a hyperparameter, that expand the resampling pool for the MBB approach (as described in section 2.2.2 (Bagging)). The orange block marks the test data for each iteration, while the gray blocks are interim points between the train and test data, reflecting the requirement from NOS of predicting two days ahead.

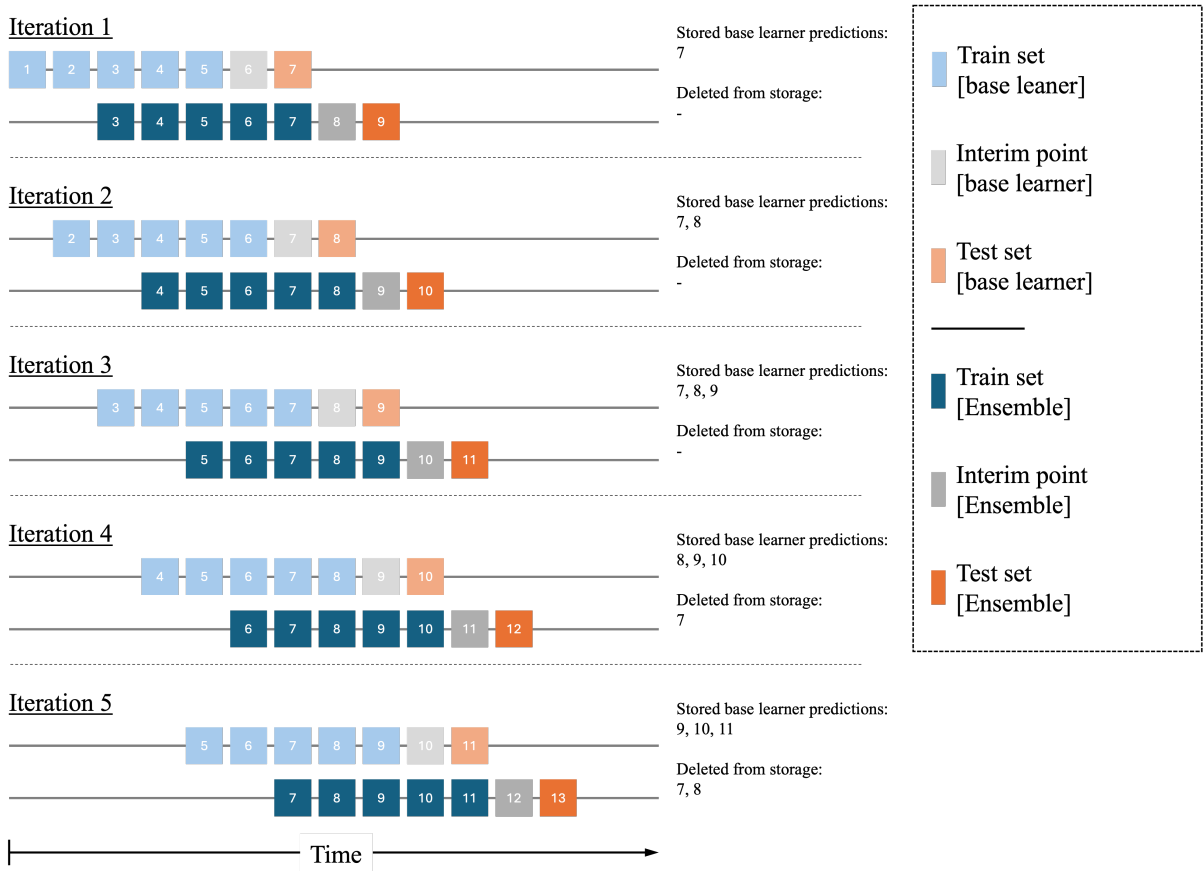


Figure A7: Example of the rolling window mechanism for DI

Figure A7 illustrates the rolling window approach as applied to DI, given an 'embed size' of three. In this setup, base-learner selection and weight allocation are based on the recent performance of each base-learner on the most recent available data, reflected in the stored predictions. Each row in the figure represents an iteration in the rolling window process, with the top row of each pair showing the predictions made by base-learners and the bottom row representing the aggregated ensemble prediction for that iteration.

Each rolling window begins with a training window for the base-learners, consisting of a fixed-length sequence of past observations, shown in light blue. Based on this sequence, each base-learner makes a prediction on the base-learner test set, shown in light orange. Between the base-learner train set and the base-learner test set, there is an interim point, reflecting the requirement from NOS of predicting two days ahead. These predictions are stored, allowing the system to track the recent performance of each base-learner. Initially, the set of stored predictions grows with each new iteration. However, once the predetermined maximum size, a hyperparameter, is reached, the oldest stored prediction is discarded each time a new prediction is added, maintaining a fixed set size that ensures only the most recent performance data is used for base-learner selection and weight allocation.

The training set for the ensemble, shown in dark blue, consists of the same number of data points as the training set for the base-learners. To ensure that the most recent data is used for the ensemble's prediction, the base-learner test set is aligned as the last point in the ensemble training set. The ensemble test point, shown in dark orange, represents the target for each prediction iteration. Using the dynamically selected base-learners and weights derived from the most recent stored predictions of the base-learners, the ensemble prediction is generated for this specific test point. After each prediction, the entire window advances forward by one step. This incremental shift enables the model to incorporate recent data and exclude older observations, adapting to potential changes in the time series.

A.5 Gridsearch Parameters

Bagging Parameters	
<i>Parameter</i>	<i>Values</i>
Embed Size:	(10), (15), (20), (10, 15), (10, 20), (10, 15, 20)
Block Size:	5, 10, 15
Number of Bootstrap Samples:	10, 50, 100
Aggregation Methods:	"mean", "median", "trimmed mean", "winsorized mean"
Additional Points:	0, 10, 20
Use All Data:	True, False
Dynamic Integration Parameters	
<i>Parameter</i>	<i>Values</i>
Embed Size:	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15
Number of Base-Learners:	2, 3, 4, 5, 6

Table A1: Grid Search Parameters for Bagging and DI

Description of each hyperparameter:

- Bagging
 - Embed size: Sizes of embedding layers.
 - Block size: Size of each block for bootstrap sampling (MBB).
 - Number of Bootstrap Samples: Number of bootstrap samples to generate.
 - Aggregation Methods: Method to aggregate predictions.
 - Additional Points: Number of additional points in the embedding pool.
 - Use All Data: Whether to use all data up to the prediction point.
- Dynamic Integration
 - Embed Size: The embed size to evaluate the base-learners before selection / weight allocation for the upcoming prediction.
 - Number of Base-Learners: The number of base-learners that make a prediction in Dynamic Weighting with Selection.

A.6 Results

A.6.1 Bagging Results

Out of 114 Pareto-efficient points in figure A8, the embed size 10 appears 64 times, 15 appears 48 times, and 20 appears 2 times:

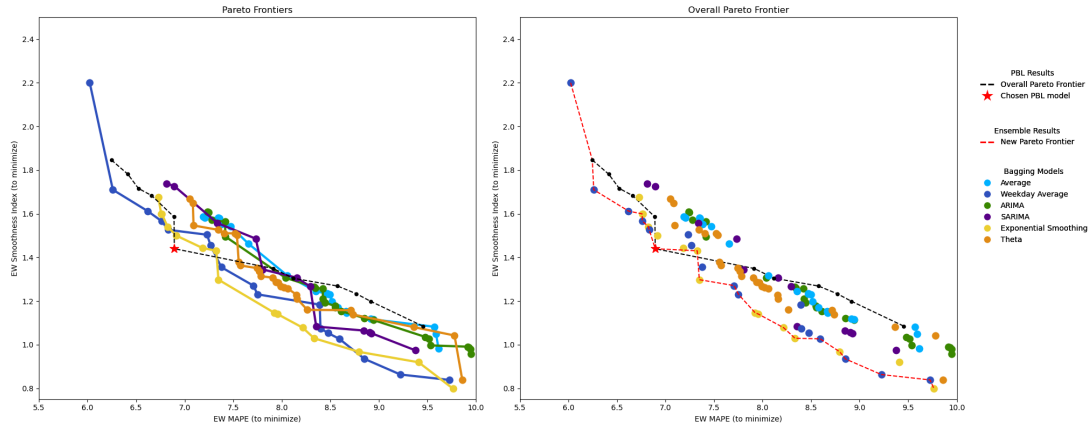


Figure A8: Pareto Frontier for Bagging given a Factor of 1.5 - single embed sizes only

Out of 93 Pareto-efficient points in figure A9, the embed size combination [10, 15] appears 55 times, while the combination [10, 20] appears 38 times:

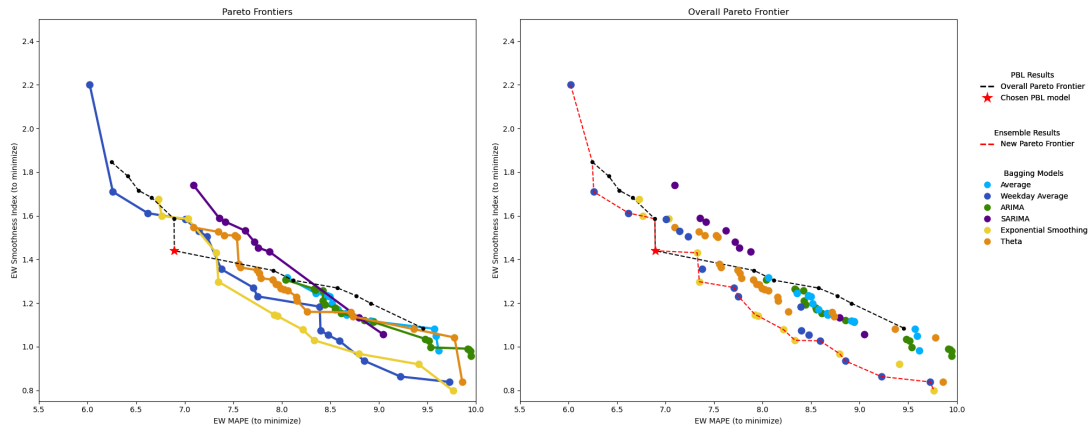


Figure A9: Pareto Frontier for Bagging given a Factor of 1.5 - multiple embed sizes only

Out of 116 Pareto-efficient points in figure A10, the embed size 10 appears 64 times, 15 appears 48 times, 20 appears once, while the combination [10, 15] appears 3 times:

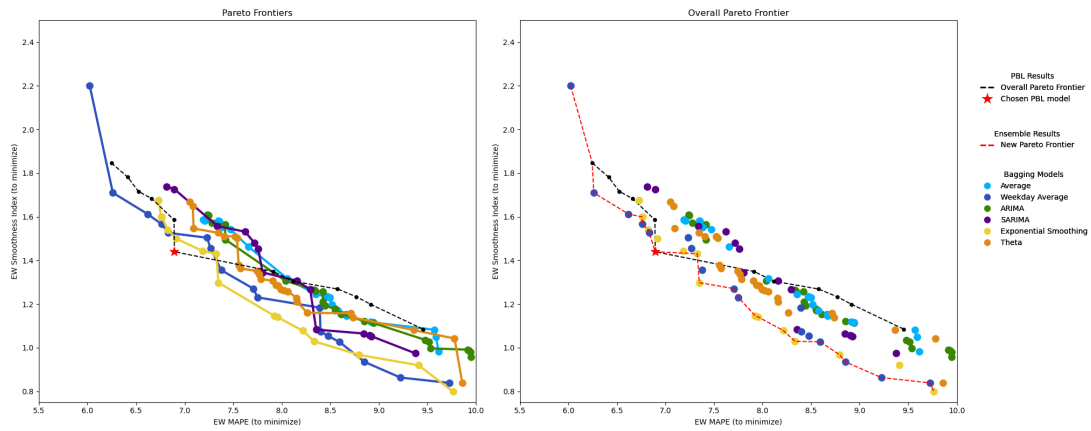


Figure A10: Pareto Frontier for Bagging given a Factor of 1.5 - all embed sizes

A.6.2 Dynamic Integration Results

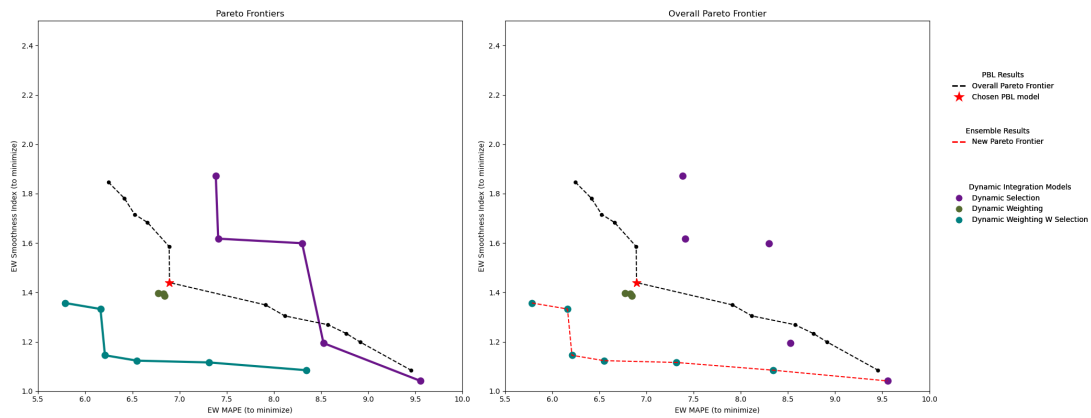


Figure A11: Pareto frontier for DI using the dynamic, error-based approach

A.6.3 Final Model Decision and Performance Evaluation

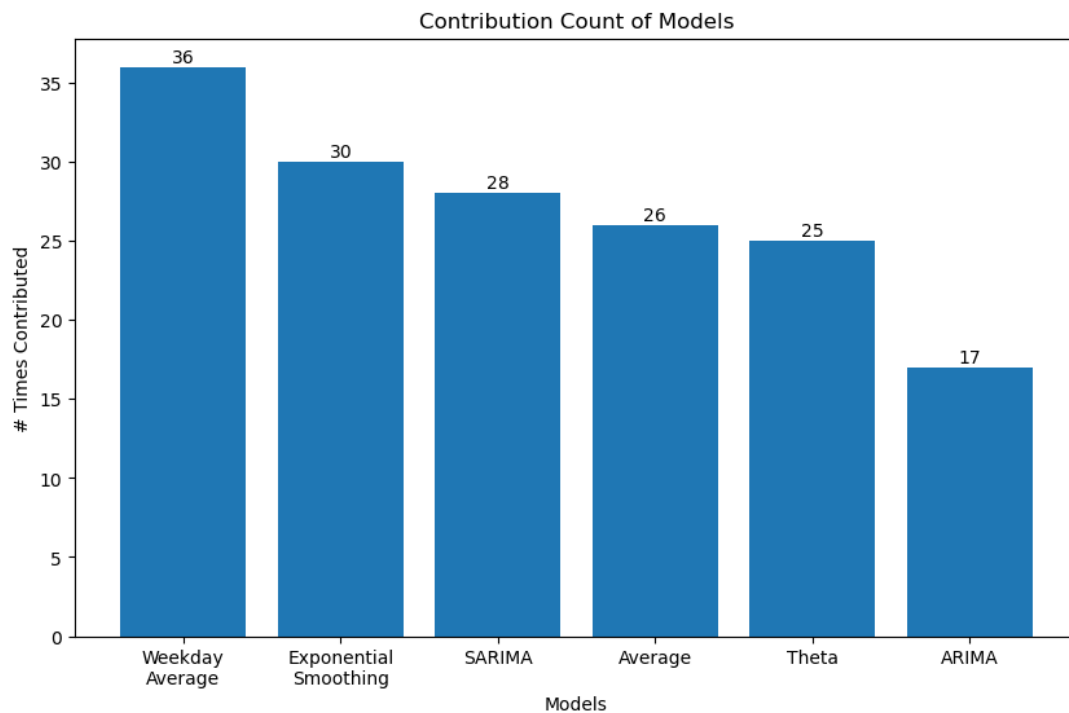


Figure A12: Contribution Frequency of Each Base-Learner Algorithm to the DWS Model

The contribution frequency of each base-learner algorithm to the DWS Model was derived by counting how often each algorithm was among the subset with the lowest prediction errors during the rolling window cross-validation process. Each time an algorithm's error falls within the select group, its performance count is incremented.

Note: Adding the #times contributed for each base-learner leads to a total of 162 contributions. This is because this configuration allows the contribution of three base-learners per prediction. Additionally, this model predicted a total of 54 days as opposed to the 40 days of the test set, a result attributable to the rolling window approach.

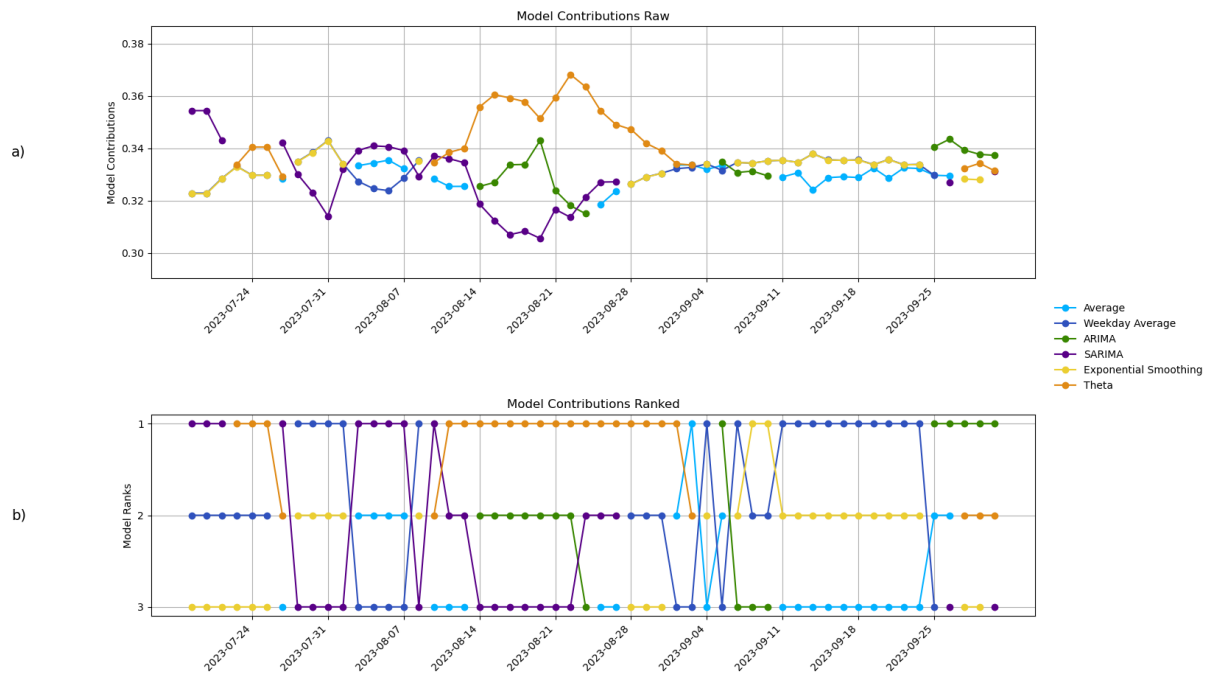


Figure A13: Contribution of Each Base-Learner Algorithm to the DWS Model over Time