Master Thesis

Improving Call Arrival Forecasting in contact centers Using Data-Driven Skill Merging and Hierarchical Time Series Forecasting

Author: Eirini Arseni (2743392)

1st supervisor: René Bekker
daily supervisor: Giuseppe Catanese (CCMath)
2nd reader: Vincent François-Lavet

A thesis submitted in fulfillment of the requirements for the VU Master of Science degree in Business Analytics

March 2026

Abstract

Accurate call arrival forecasting is central to workforce management in contact centres, but many skills operate at low volume and are hard to forecast individually. Skills are therefore often merged into groups before forecasting using manual business knowledge, but at the collaborating company this grouping is effectively one-to-one, so each skill is currently forecast on its own. This thesis investigates whether merging skills using data-driven similarity, combined with hierarchical forecasting, can improve the accuracy and coherence of forecasts compared with this current per-skill practice. Using daily call volumes for 175 skills, several time-series similarity measures and a feature-based representation were compared as inputs to three clustering algorithms, and the resulting groups were screened for validity and stability. Cluster-level Error-Trend-Seasonality (ETS) forecasts were then disaggregated to individual skills using top-down disaggregation and Minimum Trace (MinT) reconciliation.The feature-based representation produced the most stable clusters and the ones easiest to forecast, and combining it with top-down disaggregation reduced system-level forecast error by roughly 23% while keeping forecasts coherent across levels, that is, skill-level forecasts sum exactly to their cluster totals, and greatly reducing computation, since only a handful of cluster-level models are fitted instead of one model per skill. The approach also extended usable forecasts to sparse skills that cannot be modelled in isolation. MinT, by contrast, guaranteed coherence but did not improve accuracy. For this contact centre, feature-based clustering with top-down disaggregation is both the most accurate and the simplest approach. The underlying reasons, a shared weekly demand cycle that limits correlation-based grouping, and homogeneous clusters that leave little for reconciliation to exploit, are common in skill-based contact centres, suggesting the approach may transfer to similar settings, though this remains to be confirmed

Acknowledgements

Working on this thesis has been a great and genuinely interesting experience, and I am grateful to everyone who supported me along the way. First, I would like to thank my supervisor, René Bekker, for the biweekly supervision throughout the project. His inspiration, feedback, and academic guidance were invaluable and greatly shaped the direction of this work. I would also like to thank CCmath for trusting me to work with their customers' data and for offering me so much flexibility and freedom in how I approached the project. In particular, I am grateful to my daily supervisor, Giuseppe Catanese, for his guidance, practical advice, and continuous support throughout. Finally, I would like to thank my family, friends, and roommates for their encouragement, patience, and support throughout this period.

Contents

Chapter 1

Introduction

In modern service organizations, contact centers play a crucial role in handling customer interactions, ranging from technical support and sales to administrative and financial services. The performance of a contact center strongly affects both customer satisfaction and operational costs. Long waiting times, dropped calls, or poor service quality can lead to customer dissatisfaction and reputational damage, while excessive staffing increases labor costs, which form the largest cost component in most contact centers. As a result, effective workforce management (WFM) is an important operational function in contact center environments.

Workforce Management (WFM) is the process of ensuring that the appropriate number of agents are scheduled at the right times to perform the required work. The WFM cycle consists of five main steps: Forecasting, Capacity Management, Scheduling, Traffic Management, and Reporting and Analysis, and is used worldwide to plan and manage teams more effectively. Each step feeds into the next, creating a continuous flow that repeats regularly, often weekly, to adapt to changing business needs and support ongoing improvement through insights gained from each cycle.

Forecasting is the core of workforce management because all staffing and scheduling decisions are based on predicted call volumes. Errors at the forecasting stage propagate through the entire planning process: under-forecasting causes understaffing and service level violations, while over-forecasting leads to unnecessary labor costs. Because staffing decisions must be made in advance and are difficult to adjust in real time, accurate call arrival forecasting is essential for effective workforce management.

Most modern contact centers operate in a skill-based routing environment. Instead of a single queue, incoming calls are routed to different queues, also referred to as skills, on customer needs or call characteristics. Typical examples include language based queues (e.g., Dutch-speaking and English-speaking customers), service-type queues (e.g., sales versus technical support), or product-specific queues. Each skill has its own stream of incoming calls over time and therefore its own time series of call arrivals.

In practice, however, agents are often organized in departments or agent groups that handle several related skills rather than a single skill on its own. Because of this, workforce planners need accurate forecasts not only for each skill, but also for the groups of skills that are staffed together. So both individual and grouped call arrival forecasts matter for good staffing and scheduling. These staffing groups show why grouped forecasts are useful, but they do not fix

how skills should be grouped for forecasting. As argued below, the best forecasting groups are not always the organisational ones, which is why a data-driven approach is used.

However, producing accurate forecasts at multiple aggregation levels is challenging. Forecasting models generally perform better when they are trained on time series with sufficiently high volumes and stable patterns. Many individual skills in contact centers have low call volumes, irregular demand, or strong variability, which makes them difficult to forecast accurately in isolation. A common practical solution is therefore to merge multiple skills into aggregated groups and forecast their combined call volumes. These aggregated forecasts are then used as input for staffing decisions. In many organizations, such skill-merging decisions are based primarily on business knowledge and expert intuition, for example by grouping skills that belong to the same department or service type.

Grouping skills based on business knowledge is convenient, but it does not always lead to accurate forecasts. Skills that seem similar from a business point of view can have very different call patterns, while skills from different areas can behave in similar ways over time. This suggests using data-driven methods, such as time series similarity and clustering, to group skills based on how they actually behave, which can improve forecast accuracy.

At the same time, forecasting across multiple levels also creates consistency issues. Forecasts made separately for total call volume, skill groups, and individual skills may not align, leading to conflicting planning decisions. Hierarchical time series forecasting solves this by modeling the relationships between levels and adjusting forecasts so they add up correctly. Combining data-driven skill clustering with hierarchical forecasting can therefore improve both forecast accuracy and planning consistency in workforce management.

Against this background, this thesis investigates whether data-driven methods can improve call arrival forecasting in skill-based contact centers. In particular, it examines the extent to which automatically merging skills based on similarities in their call arrival patterns, using time series similarity measures and clustering techniques, can lead to more accurate forecasts than traditional business-knowledge-based aggregation. Furthermore, the study explores how hierarchical time series forecasting and reconciliation methods can be used to ensure consistency between forecasts at different aggregation levels. The relevant point of comparison is the current operational practice at the partner company, in which each skill is forecast individually with its own model and no skills are merged; this individual-skill approach therefore serves as the baseline against which the data-driven grouping strategies are evaluated throughout this thesis.

This thesis is conducted in collaboration with CCmath, a company that develops forecasting and workforce management solutions for contact centers. CCmath provides software tools, training, and consultancy services aimed at improving forecasting accuracy, staffing decisions, and scheduling efficiency. Founded by academics from Vrije Universiteit Amsterdam, the company combines scientific research, machine learning, and advanced mathematical methods with practical workforce management applications. Within CCmath's forecasting workflow, historical call arrival data are used to generate forecasts for multiple skills, which are subsequently translated into staffing requirements for operational decision-making.

The central research question of this thesis is:

Can data-driven skill-merging strategies, combined with hierarchical forecasting, im-

prove the accuracy and coherence of call arrival forecasts in contact centers compared to current workforce management practice?

To address this question, the thesis is guided by four sub-questions:

1. Which pairwise time series similarity measure best captures the dependencies between contact center skills, considering both correlation-based measures (how strongly two skills move together over time) and shape-based measures (how similar their call arrival patterns are, even when peaks occur at slightly different times)?

2. How stable are the identified skill groups across different clustering algorithms, and does the choice of similarity measure affect the robustness of the resulting groupings?

3. Does data-driven clustering improve the forecastability of aggregated skill series, measured by forecast error, compared to the current practice of forecasting each skill individually?

4. How much does hierarchical forecast reconciliation improve the coherence and accuracy of forecasts across different aggregation levels of the contact center?

The remainder of this thesis is structured as follows. Chapter 2 reviews the literature on exponential smoothing and the ETS framework, hierarchical time series forecasting and reconciliation, and identifies the research gap that motivates this study. Chapter 3 describes the dataset, the preprocessing and missing-value treatment, and the exploratory analysis of demand patterns across skills. Chapter 4 presents the methodology, covering the forecasting model, the time series similarity and feature-based distance measures, the clustering algorithms and cluster-selection criteria, the stability analysis, and the top-down and MinT reconciliation methods. Chapter 5 reports the results of the similarity, clustering, and forecasting experiments. Chapter 6 recaps the key findings and discusses their business implications for CCmath, and Chapter 7 concludes by answering the research questions and outlining directions for future work.

Chapter 2

Literature Review

Call arrival data in contact centers has several characteristics that directly shape the choice of forecasting method. Arrivals are observed at high frequency and show multiple seasonal patterns at the same time, demand changes strongly both within the day and across the week, so that intraweek seasonality must be modelled explicitly [12]. Because many individual skill queues operate at low volume, sparse or irregular series are common, and these are hard to forecast accurately on their own [15]. This motivates aggregation strategies and hierarchical approaches that pool information across related series. A further consistent finding is that call arrivals are overdispersed relative to the Poisson benchmark, driven by burstiness, shared external demand drivers, and time-varying arrival rates, which means that flexible continuous approximation models are generally preferred over strict count-based specifications [11].

This chapter reviews the literature relevant to these characteristics. Section 2.1 examines exponential smoothing methods and why the ETS framework is particularly well-suited to this setting. Section 2.2 reviews hierarchical time series forecasting and reconciliation methods, drawing on evidence from adjacent fields to motivate their use here. Section 2.3 concludes by identifying the research gap and the contribution of this thesis.

2.1 Exponential Smoothing and the ETS Framework

Exponential smoothing is one of the most widely used families of time series forecasting methods in operational settings. The general framework, known as Error-Trend-Seasonality (ETS) modelling, was formally brought together by [9] into a state space formulation that allows systematic model selection and principled uncertainty quantification. Each ETS model breaks the forecasting problem down into three structural components: error, trend, and seasonality, each of which can be specified as additive, multiplicative, or absent, giving a broad family of models that can be matched to the structural characteristics of any given time series.

The appeal of ETS for operational demand forecasting rests on several properties. First, exponential smoothing gives exponentially decaying weights to past observations, so that recent demand levels receive more weight than older ones. This is particularly well suited to contact center environments, where demand patterns shift gradually over time due to changes in the customer base, service setup, or external conditions, and where recent behaviour is generally more useful for predicting near-future demand than observations from several months ago [7].

Second, ETS models are parsimonious: they are described by a small number of smoothing parameters that can be estimated efficiently from moderate amounts of data, making them practical for production systems that need to fit and maintain hundreds of models at the same time. Third, the Holt-Winters seasonal extension of this framework explicitly models trend and seasonal components through separate updating equations, with the seasonal period set to match the dominant cycle in the data. In contact center demand, day-of-week seasonality is consistently the strongest structural driver across the literature [12, 28], and a weekly seasonal period of seven directly captures this structure.

In line with these properties, [28] shows that exponential smoothing methods with appropriate seasonal specifications perform as well as or better than more complex statistical models for intraday call arrival forecasting, while being much simpler to implement and maintain in operational pipelines. This finding is consistent with the broader forecasting literature: in the M3 and M4 forecasting competitions, which compare a wide range of methods across thousands of real-world time series, ETS consistently ranks among the most accurate general-purpose methods, particularly for series with strong seasonal patterns and moderate amounts of training data [18, 7].

The choice between additive and multiplicative specifications depends on whether seasonal fluctuation size is roughly constant over time or scales with the level of the series. For contact center skills, demand levels can change substantially across periods due to long-term growth or decline, and multiplicative seasonality is often more suitable where fluctuation size grows with volume. A practical and widely used approach is to apply a logarithmic transformation to the series, which turns multiplicative structure into an additive one on the transformed scale and also stops the model from producing negative forecasts for count data that is always non-negative [7]. This approach works well for low-volume skills with near-zero observations, which are common in skill-based contact center environments.

Compared to other model families, ETS has clear practical advantages in this setting. ARIMA models require the analyst to identify the autoregressive and moving-average orders and to apply stationarity transformations such as differencing before fitting, that is, to choose how many past values and past errors the model should depend on and to remove trends so that the statistical properties of the series stay stable over time. Doing this reliably for hundreds of skills is difficult to automate. Machine learning approaches such as gradient boosting or recurrent neural networks can capture complex nonlinear patterns but need much more data, careful hyperparameter tuning, and produce forecasts that are harder to interpret or check in a production workflow. ETS models, by contrast, are robust to moderate deviations from their assumptions, fast to estimate, and produce interpretable forecasts with well-calibrated prediction intervals [9]. For these reasons, ETS-based forecasting has become a standard baseline and production choice in operational demand forecasting across industries including retail, healthcare, and telecommunications [18].

2.2 Hierarchical Time Series Forecasting and Reconciliation

Many practical forecasting problems require consistent predictions at multiple levels of aggregation at the same time. In retail, forecasts are needed at the product, store, and regional level.

In energy systems, demand must be predicted at the household, district, and national level. In contact centers, call arrival forecasts are needed both at the individual skill level and at the level of grouped skills used to drive staffing decisions. When forecasts are produced independently at each level, they are generally not consistent: skill-level forecasts do not add up to the corresponding group-level forecasts, leading to conflicting planning signals. Hierarchical time series forecasting addresses this by modelling the relationships between levels and adjusting forecasts so that they satisfy the aggregation constraints of the hierarchy.

The core framework for hierarchical forecasting and reconciliation was established by [10], who showed that any linear reconciliation of base forecasts can be written as a projection matrix applied to the full vector of forecasts across all levels. The optimal reconciliation under this framework minimises the trace of the covariance matrix of reconciled forecast errors, giving the Minimum Trace (MinT) estimator formalised by [30]. MinT is theoretically attractive because it produces unbiased reconciled forecasts with minimum variance, making use of the full cross-series error covariance structure to redistribute forecast error across the hierarchy. The practical difficulty is that the covariance matrix must be estimated from in-sample residuals, and this estimation becomes unreliable when the number of series is large relative to the length of the training period. Shrinkage-based regularisation, such as the Ledoit-Wolf estimator, helps by improving the conditioning of the covariance estimate [30].

A simpler and historically common alternative is top-down disaggregation, in which forecasts are produced at an aggregate level and then split across lower-level series using fixed proportions taken from historical data [6]. Top-down approaches are computationally efficient and benefit directly from the pooling effect of forecasting aggregate series: adding up many individual series smooths out random noise and makes trend and seasonal signals clearer for the forecasting model. The main drawback is that top-down methods throw away information about the individual behaviour of lower-level series and may allocate demand incorrectly when historical proportions are not stable over time [1].

Evidence from several fields shows that hierarchical forecasting outperforms independent single-level approaches. In retail demand forecasting, [1] showed that hierarchical reconciliation improves consistency and reduces forecast error at multiple aggregation levels compared to forecasts produced independently. In energy demand forecasting, [8] showed that reconciliation produces meaningful accuracy gains at lower levels of the hierarchy where individual series are noisy and have few observations. In tourism forecasting, hierarchical methods consistently outperformed both bottom-up and top-down approaches when lower-level series are volatile [1]. In public health, reconciled hierarchical forecasts of disease incidence have been shown to improve local-level predictions by borrowing strength from higher-level aggregate signals [21]. Across all these fields the underlying mechanism is the same: reconciliation lets information from stable, high-volume aggregate series flow down to noisy, low-volume component series, improving forecast quality precisely where independent modelling performs worst.

The parallels with the contact center setting are clear. Individual skill queues in skill-based routing environments are exactly the kind of low-volume, high-variability series that benefit most from hierarchical pooling. Grouping related skills and forecasting at the cluster level reduces noise and gives the ETS model a cleaner signal to work with, while top-down disaggregation or

MinT reconciliation recovers skill-level forecasts that are consistent with operational planning constraints. This combination of aggregation and reconciliation has not, to the best of the author's knowledge, been studied in the contact center forecasting literature, where existing work largely treats forecasting as a single-level task without ensuring consistency across levels.

2.3 Research Gap and Contribution

The literature reviewed above points to a clear gap at the intersection of contact center forecasting and hierarchical time series methods. On the forecasting side, exponential smoothing methods have been shown to suit the seasonal and operational characteristics of call arrival data well [28, 12]. On the hierarchical forecasting side, reconciliation methods such as MinT have been shown to improve consistency and reduce error in multi-level forecasting settings across retail, energy, tourism, and healthcare. However, their application to the specific structure of skill-based contact centers, where the hierarchy is built from data-driven clustering rather than a fixed organisational tree, has not been explored.

A second gap concerns how skills should be grouped before hierarchical forecasting is applied. Current practice in workforce management relies on business-knowledge groupings by language, product type, or department. While operationally intuitive, these groupings do not guarantee that the resulting aggregated series are easy to forecast accurately. Recent work in retail demand forecasting [27] and sparse time series [15] suggests that grouping series by behavioural similarity before forecasting can substantially improve accuracy by making sure that pooled series share the same demand dynamics. No comparable study has applied data-driven skill grouping in a contact center hierarchical forecasting context.

This thesis addresses both gaps together. It investigates whether grouping skills by data-driven similarity produces more forecastable aggregates than standard business-knowledge groupings, and examines how hierarchical reconciliation can be applied on top of these groupings to produce forecasts that are both accurate and consistent across operational planning levels. These two gaps directly motivate the research question and sub-questions stated in Chapter 1; the specific methods used to address them are introduced and formally defined in Chapter 4.

Chapter 3

Data Description and Preprocessing

This chapter describes the data used in the thesis and the steps taken to prepare it for analysis. Section 3.1 introduces the dataset and its main characteristics. Section 3.2 describes the cleaning and preprocessing pipeline, including the handling of missing values. Section 3.3 then presents an exploratory analysis of the demand patterns across skills, which informs the modelling choices made in later chapters.

3.1 Dataset description

The dataset used in this study originates from one of CCmath's customers. Due to confidentiality agreements, the name of the customer and the specific skill names are not disclosed in this report. The dataset contains historical call arrival information extracted from the customer's telephony platform and integrated into CCmath's forecasting tool. In other words, the dataset contains only contact volumes, the dates and skill names.

Data availability varies across skills. For a subset of skills, historical data is available from 2020 onward, while for the majority of skills, reliable data is available starting from the middle of 2023 until the middle of 2026. For this study, the dataset spans from September 2023 to the end of June 2026. The raw data is recorded at a 15-minute granularity, which is the most detailed level available within the forecasting tool. For this project, however, the analysis is performed on daily aggregated call volumes rather than 15-minute data. This decision was made to reduce computational complexity, accelerate experimentation, and simplify exploratory analysis. Using daily data enables faster clustering and forecasting model development, while still preserving the main demand patterns required for the methodological evaluation. Higher-frequency data can be incorporated in future work, once the methods have been validated at the daily level, since intraday forecasts are ultimately required for operational staffing and scheduling. The system also supports aggregation to higher time levels, including hourly, daily, weekly, monthly, and yearly intervals.

In the context of contact center operations, a skill represents a specific queue or call line, typically associated with a particular service type, language, or customer request category. The customer dataset initially contains 275 skills, providing a sufficiently large set of time series to support clustering and hierarchical forecasting analysis.

The raw data collected from the telephony platform includes information such as skill iden-

tifier, date, time, and call volume per interval. Within the forecasting tool, these records are aggregated by summing call volumes per skill and time interval. The processed dataset therefore consists of time series of call volumes indexed by skill, date, and time. The tool also allows exporting the data for external analysis.

3.2 Data cleaning and preprocessing

The data preprocessing stage was designed to ensure data quality, consistency, and suitability for time series clustering and forecasting. The preprocessing pipeline consists of data standardization, missing value handling, skill filtering, and time coverage validation. An initial exploratory data analysis (EDA) was performed to assess overall data completeness and structure. The dataset contains 269,478 total observations across 275 unique skills, of which 176,892 (65.64%) contained valid numeric values and 92,586 (34.36%) were missing, represented by the symbol "-". No days were missing from the calendar: every date in the period was present and parseable, so the missing values reported above concern dates that have no recorded call volume rather than gaps in the date sequence. Two skills contained only missing values and were removed from further analysis, leaving 273 skills.

During standardization, the date variable was converted to a standard datetime format and all call volume entries were converted to numeric format. Missing observations represented by "-" or empty strings were systematically converted to NaN to ensure compatibility with statistical and machine learning methods. A clear distinction was maintained throughout between true zero demand, indicating that no calls arrived during a valid observation interval, and missing values, indicating that no information was recorded for that interval. To ensure reliable clustering and forecasting, skills with insufficient data coverage were subsequently removed in four stages, summarised in Table 3.1.

Filter step	Skills removed	Skills remaining
Initial dataset	—	275
Remove fully missing skills	2	273
Remove dead skills	26	247
Remove truly new skills	8	239
Remove sparse skills ($> 50\%$ missing)	64	175

Table 3.1: Summary of skill filtering steps applied during preprocessing.

First, skills containing only missing values were removed, as they provide no usable information for modeling. Two skills were removed during this step, leaving 273 skills. Second, skills were classified as dead or truly new based on their temporal activity relative to the end of the dataset, using two cutoffs. A recency cutoff, defined as two months before the final observation date, was used to identify dead skills, while a one-year cutoff, defined as twelve months before the final observation date, was used to identify truly new skills. These thresholds distinguish between skills that are still operationally active and those that have either been discontinued or only recently introduced. Dead skills are defined as skills whose last observed data point falls

on or before the recency cutoff, indicating that the skill is no longer active. These skills were excluded because including discontinued skills would introduce time series that do not reflect current operational behaviour, potentially distorting clustering and forecasting results. Truly new skills are defined as skills whose first observed data point falls after the one-year cutoff, and which are not already classified as dead, meaning they only appear within the final year of the dataset. These skills were excluded because they lack sufficient historical depth to support reliable time series modeling and pattern recognition. As shown in Figure 3.1, 26 dead skills and 8 truly new skills were identified and removed.

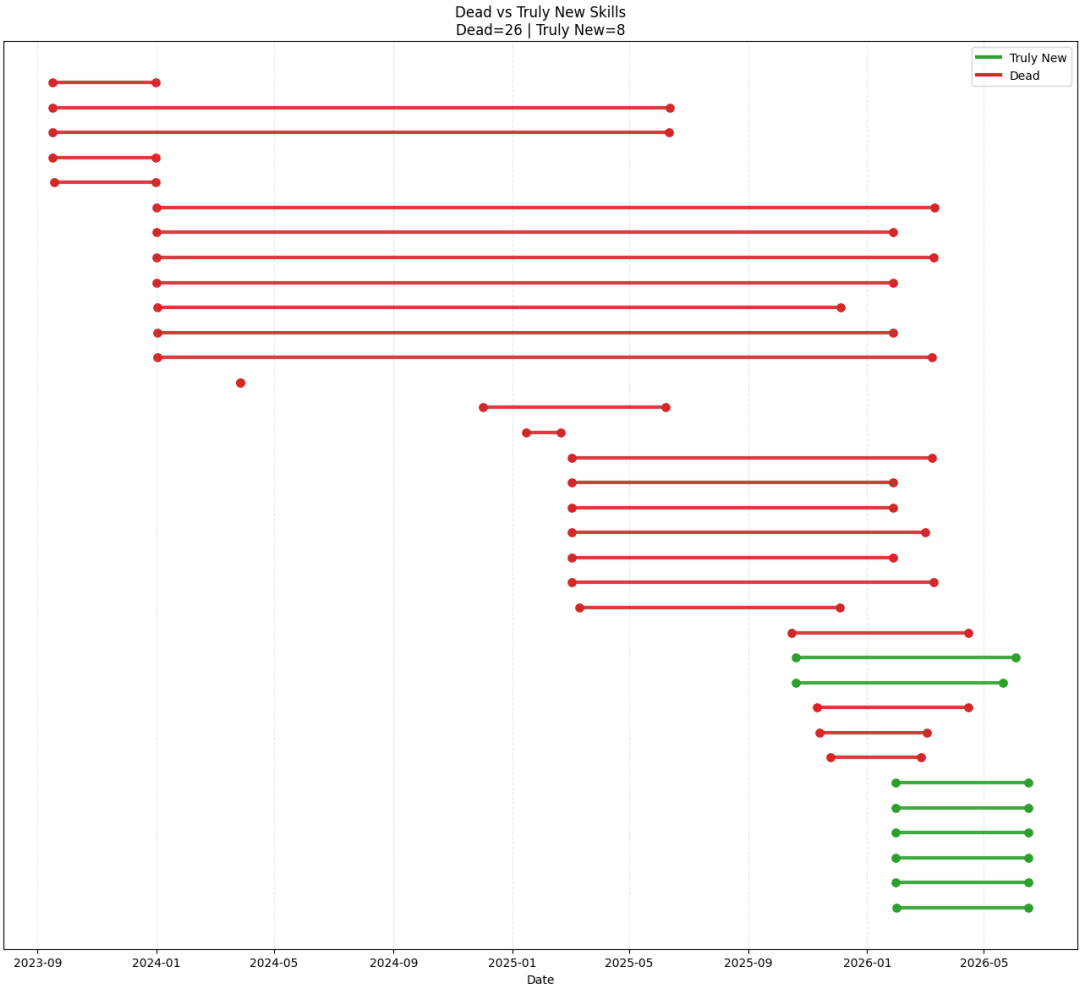


Figure 3.1: Temporal lifespan of dead and truly new skills. Red lines represent dead skills, whose last observation falls on or before the recency cutoff (two months before the final observation date). Green lines represent truly new skills, whose first observation falls after the one-year cutoff (twelve months before the final observation date). 26 dead skills and 8 truly new skills were identified and removed from the analysis.

Third, a missing data ratio was calculated per skill across the full observation period. Skills with more than 50% missing observations were classified as sparse and excluded from the analysis. This threshold was selected to ensure that retained skills contain enough observed data points to support time series modeling. Skills already classified as dead were excluded from this check. In total, 66 skills exceeded the 50% missing threshold; of these, 2 had already been removed as truly new, so 64 additional skills were removed at this step. Figure 3.2 compares

the monthly missing percentage of the 64 sparse skills removed at this step with that of the 175 established skills retained for analysis. The two groups are clearly separated throughout the whole period. The sparse skills have a consistently high missing rate, remaining above 80% for most months from late 2023 through early 2025, before dropping sharply to around 58% in early 2025 and then fluctuating between roughly 46% and 60%, ending near 57% by mid-2026. The established skills, by contrast, start at around 25–30% missing in late 2023, fall to roughly 10% from early 2024 onward, and remain low and stable for the rest of the period, rising only mildly to around 15–20% in 2026. This large and persistent gap confirms that the sparse skills do not provide sufficient data coverage for reliable modeling, whereas the established skills are well populated throughout.

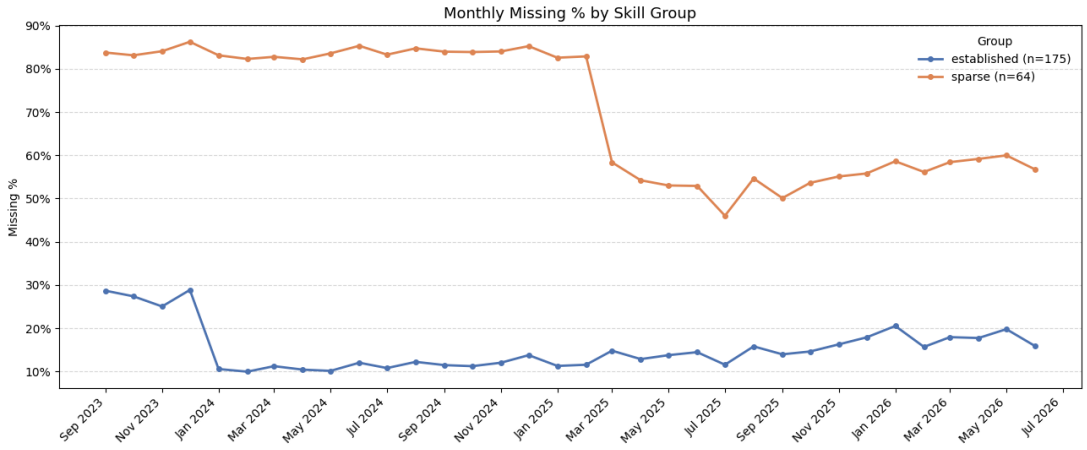


Figure 3.2: Monthly missing percentage for the 64 sparse skills removed during filtering (orange) and the 175 established skills retained for analysis (blue). The sparse skills remain above 80% missing for most of the earlier period before declining to around 58% in early 2025 and fluctuating between roughly 46% and 60% thereafter, while the established skills stay close to 10% from early 2024 onward. The persistent gap confirms that the sparse skills lack sufficient coverage for reliable modeling.

After all filtering steps, 175 skills remained for further analysis, as visualised in Figure 3.3. The coverage plot shows that the majority of skills contain data starting from September 2023, while a smaller subset begins recording from around early 2024. Nearly all skills extend through mid-2026, indicating broad coverage across the full analysis period. The uneven historical start dates across skills create some variation in series length, but all retained skills contain sufficient observations for time series modeling.

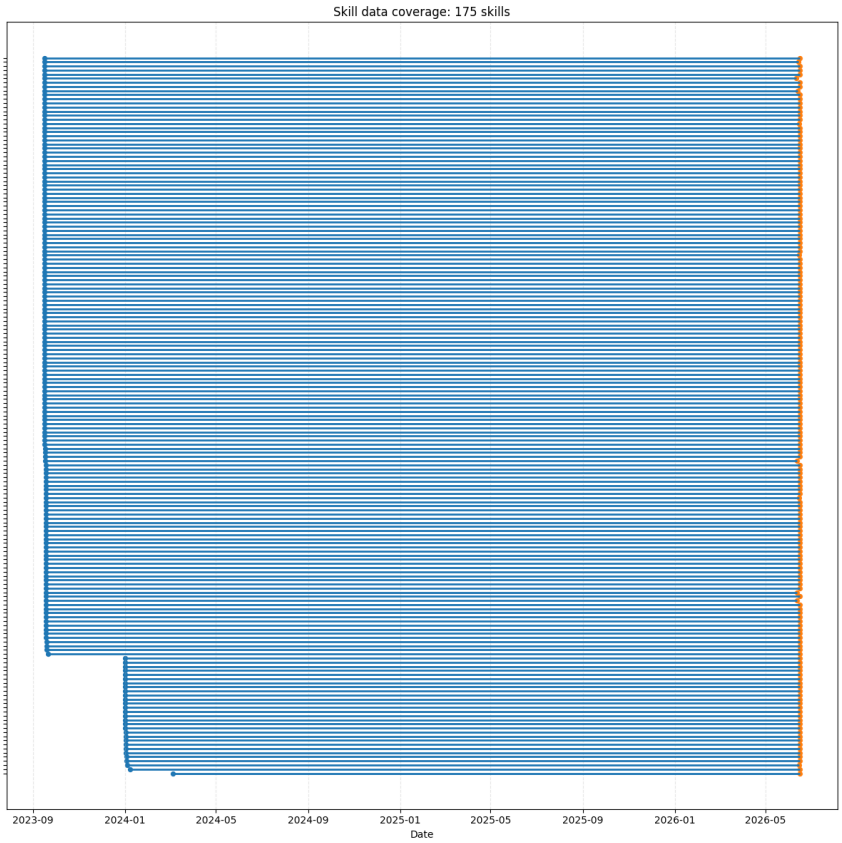


Figure 3.3: Skill data temporal coverage after all filtering steps. Each horizontal line represents one skill, where the left endpoint indicates the first available observation and the right endpoint indicates the last available observation. The majority of the 175 remaining skills contain data from September 2023 through mid-2026, with a subset beginning around early 2024.

3.2.1 Weekend Zero Imputation

A targeted imputation strategy was applied to address cases where weekend observations were systematically missing for specific skills. For each skill, the proportion of missing observations was calculated separately for Saturdays and Sundays. If more than 75% of observations for a given weekend day were missing across the full observation period, all remaining missing values on that day were imputed as zero. This threshold was chosen to distinguish between skills that genuinely do not operate on weekends and skills that simply have occasional missing weekend observations. A missing rate above 75% for a specific day of the week is unlikely to reflect random data collection issues and instead strongly suggests that the skill is structurally inactive on that day. Imputing these values as zero is therefore consistent with the expected operational behaviour of the skill rather than an arbitrary substitution. Applying this rule affected 47 out of 175 skills and resulted in the imputation of 11,081 missing observations with zero values. The remaining missing observations after this step were addressed in the subsequent imputation stage.

3.2.2 ETS-Based Imputation of Remaining Missing Values

After applying targeted structural imputation for systematic missing weekend values, some missing observations still remained across the 175 retained skills. These were assumed to primarily reflect data collection gaps rather than true zero demand, and were imputed using a per-skill Exponential Smoothing (ETS) model.

For each skill, the observed series was first pre-filled to obtain a complete sequence suitable for model fitting: interior gaps were closed by linear interpolation, and any leading or trailing missing values were resolved by forward and backward filling, respectively. This pre-filled series was used exclusively for model estimation and did not replace original observations. Importantly, ETS models were fitted exclusively on the first 70% of each skill's observations, leaving the remaining 30% untouched during model estimation. This ensures that the test portion of the data is not influenced by imputation derived from future observations, preserving the integrity of downstream forecast evaluation.

Two candidate ETS configurations were evaluated for each skill: an additive trend with additive seasonal component, and an additive trend with multiplicative seasonal component. Both models assumed a weekly seasonal cycle of seven periods, consistent with the day-of-week demand structure characteristic of call centre operations. Model parameters were estimated automatically from the training portion of the data for each skill, and the configuration with the lower AIC was selected; because the smoothing parameters are fitted per skill, they are not reported individually. All 175 skills were best described by an additive trend with multiplicative seasonal component. If ETS estimation failed entirely, missing values were filled using a centred seven-day rolling median, with any remaining gaps resolved by the series median and zeros as a last resort. A skill-specific cap was applied to all imputed values, defined as twice the 99th percentile of each skill's observed demand prior to imputation, and all imputed values were clipped to be non-negative.

After imputation, no missing values remained in the training portion of the dataset. The final dataset contains 175,875 observations in total, of which 149,175 (84.82%) correspond to originally observed values. The remaining 26,700 observations (15.18%) were not originally observed. Of these, 21,109 values were imputed in the training portion, while 5,591 missing values in the test portion were intentionally left unchanged for evaluation purposes and handled separately. The test portion contains 52,130 real observations, which were retained for forecast evaluation.

To preserve the distinction between observed and imputed values for downstream evaluation, a boolean flag *is_real* was saved for each observation, ensuring that forecast evaluation metrics can be computed exclusively on real observations. Figure 3.4 shows the monthly count of observed and imputed values across the full analysis period. Observed values consistently form the majority of data points in each month, confirming that imputation does not dominate the dataset. Figure 3.5 shows the seven-day rolling mean of total call volume before and after imputation. A visible gap is present in the early part of the observation period, reflecting the higher proportion of missing observations at that time. From early 2024 onward the lines closely overlap, indicating that imputation has a limited effect on the aggregate demand signal and does not distort the underlying temporal structure of the data.

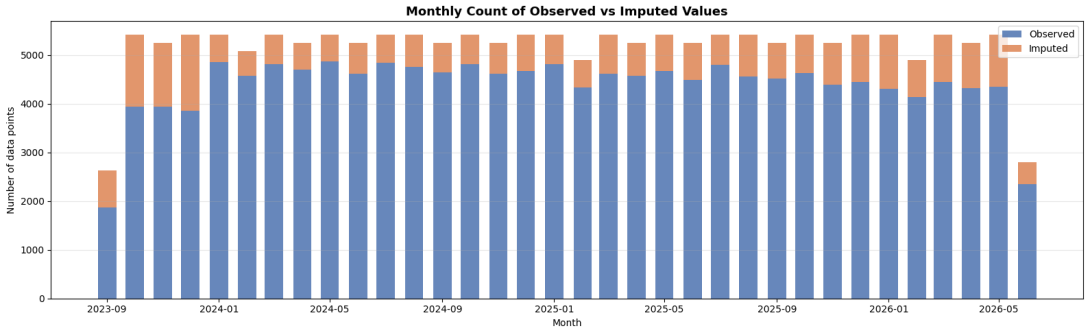


Figure 3.4: Monthly count of observed and imputed values across the full analysis period. Blue bars represent originally observed values, while orange bars represent imputed values. Observed values form the majority in each month throughout the analysis period.

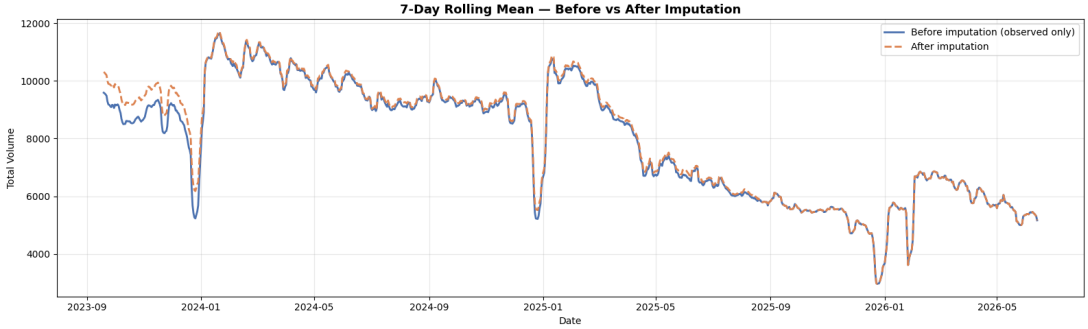


Figure 3.5: Seven-day rolling mean of total call volume before and after imputation. The gap between the two lines in the early part of the observation period reflects the higher proportion of missing observations at that time. From early 2024 onward the lines closely overlap, indicating limited impact of imputation on the aggregate demand signal.

3.3 Exploratory Data Analysis

Following data preprocessing and missing value treatment, an exploratory data analysis (EDA) was conducted to better understand the temporal structure and distributional characteristics of call demand across skills. The objective of this analysis is to identify key demand patterns, including seasonality, trend behaviour, and demand concentration across calendar dimensions. Understanding these patterns is essential for informing subsequent clustering and forecasting model design.

The EDA focuses on aggregated demand behaviour across the full set of 175 retained skills to identify system-level characteristics. While skill-level variability exists, analysing aggregated demand provides a robust overview of global demand dynamics and allows identification of dominant temporal patterns that influence workforce planning and forecasting performance.

3.3.1 Aggregate Demand Patterns

Figure 3.6 presents aggregated call volume distributions by year, month, and day of week using the fully reconstructed dataset after imputation. The yearly distribution shows that 2024 recorded the highest total call volume at 3,574,588 contacts (44.2%), followed by 2025 with

2,547,077 contacts (31.5%), 2023 with 979,652 contacts (12.1%), and 2026 with 978,446 contacts (12.1%). The comparatively low totals for 2023 and 2026 reflect the partial calendar coverage of the dataset in those years, which begins in September 2023 and extends only through June 2026. Because both 2024 and 2025 are full years, the drop from 2024 to 2025 is a real fall in call volume, not just an effect of partial coverage like in 2023 and 2026. It fits the long-term downward trend seen in the daily series (Figure 3.6) and is mainly explained by the contact centre automating some skills, so that calls which used to be handled by agents are now handled by AI. The monthly distribution reveals a clear calendar pattern, with January recording the highest total volume at 809,261 contacts (10.0%), followed by March with 795,405 contacts (9.8%) and February with 789,498 contacts (9.8%). The lowest monthly volumes occur in August, July, and June, with 467,310 contacts (5.8%), 494,365 contacts (6.1%), and 575,583 contacts (7.1%), respectively. A secondary increase is visible in the autumn, with October and November accounting for 757,065 contacts (9.4%) and 715,826 contacts (8.9%). This suggests stronger demand at the beginning of the year and again in autumn, with reduced demand during the summer months.

The day-of-week distribution confirms a strong intraweek pattern. Tuesday records the highest total volume with 1,477,699 contacts (18.3%), followed by Wednesday with 1,427,383 contacts (17.7%), Monday with 1,406,006 contacts (17.4%), and Thursday with 1,380,546 contacts (17.1%). Friday is lower at 1,208,536 contacts (15.0%), while Saturday and Sunday show substantially reduced demand, with 603,953 contacts (7.5%) and 575,640 contacts (7.1%), respectively. This strong weekday-weekend contrast is consistent with the operational structure of the skills in this dataset, many of which are inactive or reduced-capacity on weekends.

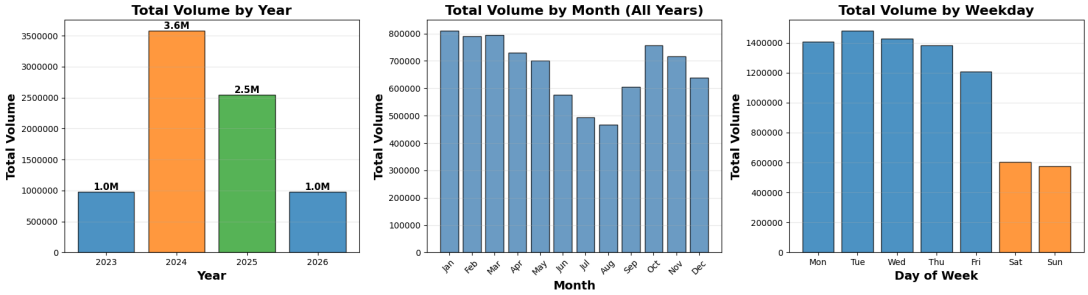


Figure 3.6: Aggregated call volume distribution across calendar dimensions. Total demand by year is shown on the left, demand by month aggregated across years is shown in the middle, and demand by day of week is shown on the right. The distributions are based on the final dataset after missing value imputation.

Figure 3.7 shows the total call volume per day across the full observation period from September 2023 to June 2026. The series exhibits a clear and persistent weekly cycle throughout the entire period, with regular peaks and troughs corresponding to weekday and weekend demand patterns, respectively. A long-term downward trend is visible, with daily volumes declining from peaks of approximately 12,000–15,000 calls during the high-volume period in 2024 and early 2025 to approximately 2,500–9,000 calls by 2026. Several sharp temporary drops are visible around year-end periods, followed by rapid recoveries. A notable spike occurs around January 2025, where daily volume briefly exceeds the surrounding demand levels before returning to the declining trend. Overall, the figure confirms both strong weekly seasonality and a substantial

reduction in aggregate call volume over time.

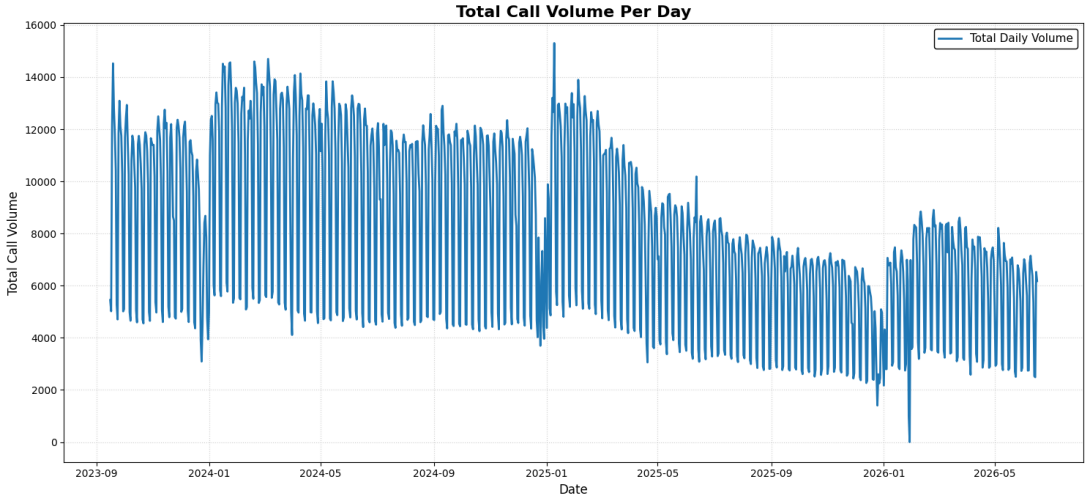


Figure 3.7: Total call volume per day across the full observation period from September 2023 to June 2026. The series shows a persistent weekly seasonal cycle and a long-term declining trend in aggregate demand.

Figure 3.8 presents the system-wide daily mean and median call volumes per skill, together with their seven-day rolling counterparts. The persistent gap between the rolling mean and rolling median throughout the observation period reflects the right-skewed distribution of skill-level demand: a relatively small number of high-volume skills substantially inflate the mean relative to the median. The rolling median remains much lower than the rolling mean across the full period, indicating that the typical skill operates at a low to moderate volume level. The rolling mean declines over time, consistent with the aggregate downward trend observed in the total daily volume series. The high-frequency oscillations in both the mean and median series reflect the weekly demand cycle operating at the skill level.

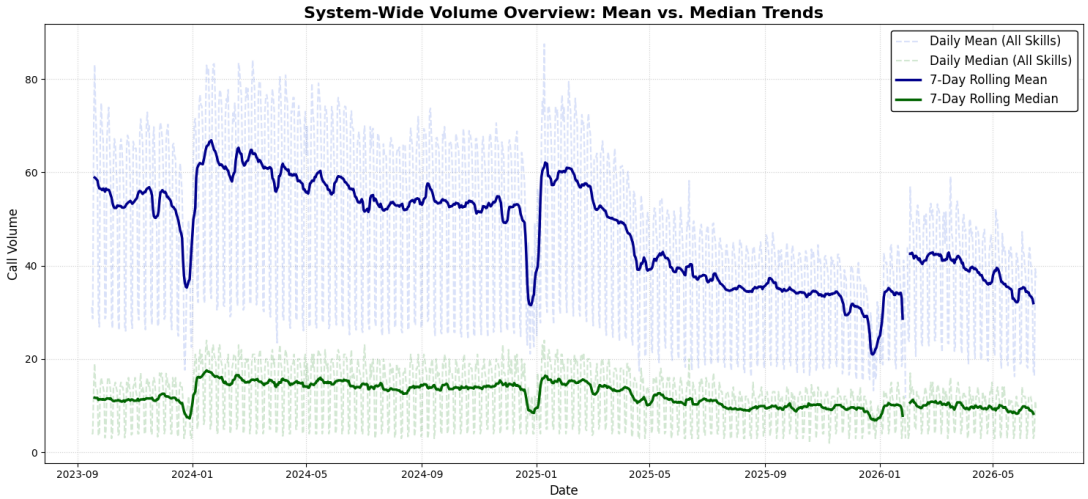


Figure 3.8: System-wide daily mean and median call volumes per skill, with seven-day rolling averages. The persistent gap between mean and median reflects a right-skewed demand distribution driven by a small number of high-volume skills. Both series exhibit a long-term declining trend and strong weekly seasonality.

3.3.2 Skill-Level Demand Distribution

Figure 3.9 presents the distribution of mean daily call volume across the 175 retained skills. The distribution is strongly right-skewed, with a mean of 46.43 calls per day and a median of 13.79 calls per day. This confirms that most skills operate at relatively low volume, while a small number of high-volume skills drive a large share of aggregate demand. The minimum mean daily volume is 1.68 calls, while the maximum is 544.44 calls, further highlighting the strong heterogeneity across skills. On the log scale, the distribution is more spread out and easier to interpret, showing that the skill portfolio contains several distinct volume ranges rather than a homogeneous group of comparable series. This heterogeneity has direct implications for forecasting: low-volume skills are inherently more difficult to forecast accurately in isolation, which motivates the use of clustering and aggregation strategies.

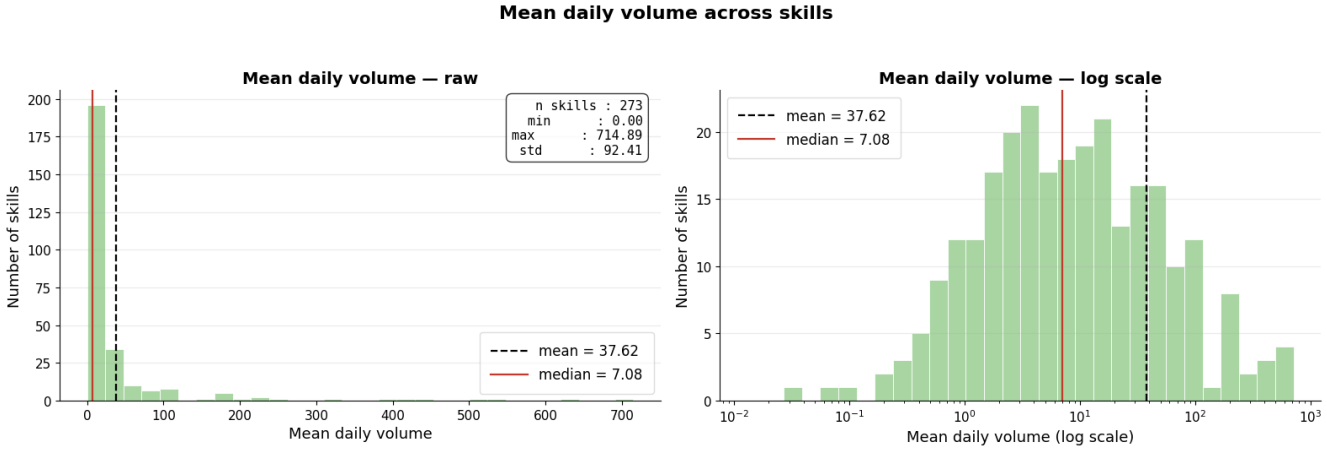


Figure 3.9: Distribution of mean daily call volume across 175 retained skills on the raw scale and log scale. The distribution is strongly right-skewed on the raw scale, with a mean of 46.43 calls per day per skill and a median of 13.79 calls per day per skill.

A key property of call arrival data is overdispersion relative to the Poisson benchmark. Under a Poisson process, the variance-to-mean ratio (VMR) equals 1. Figure 3.10 shows that the VMR distribution across the 175 retained skills has a median of 7.10 and a mean of 17.74, both substantially above this benchmark. The VMR ranges from 0.95 to 216.37, confirming widespread overdispersion across the skill portfolio. This is consistent with the well-established finding that contact center arrivals often exhibit greater variability than a simple Poisson process would predict, due to burstiness, shared demand drivers, and time-varying arrival rates. Because this ratio is computed on the raw series, it also reflects the systematic trend and day-of-week variation rather than random noise alone, so it overstates the true overdispersion; after accounting for this time structure the residual noise may be closer to Poisson. The ratios are nonetheless large enough that some genuine overdispersion likely remains, which motivates the use of flexible forecasting approaches rather than a standard Poisson specification.

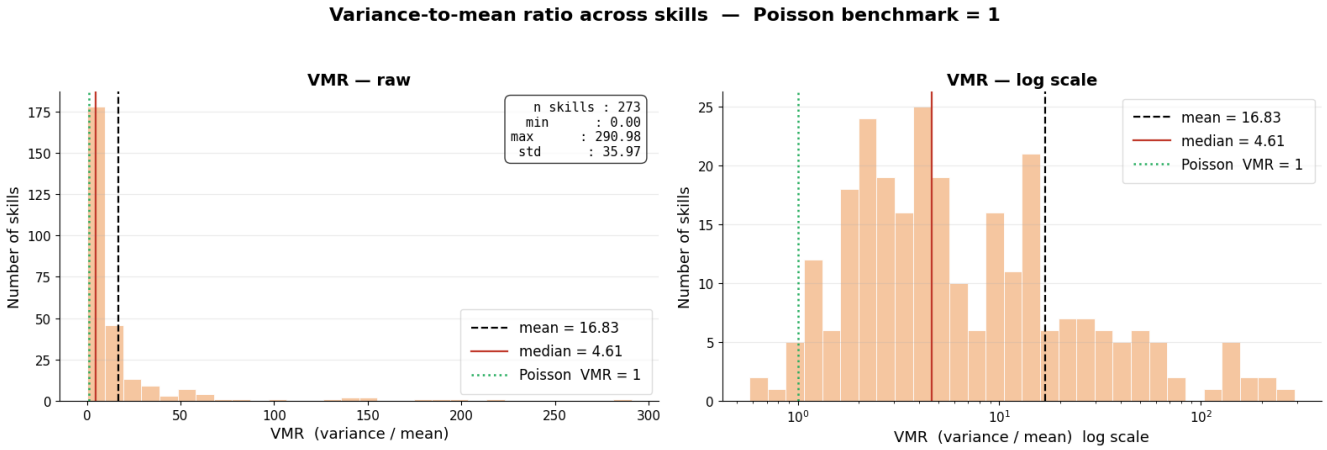


Figure 3.10: Distribution of the variance-to-mean ratio (VMR) across 175 retained skills on the raw scale and log scale. The Poisson benchmark of VMR = 1 is shown as a dotted green line. The median VMR of 7.10 and mean VMR of 17.74 confirm widespread overdispersion across skills.

3.3.3 Seasonality Analysis

To characterise the seasonal structure of each skill, a one-way ANOVA was conducted separately for day of week (DOW), day of month (DOM), and month of year (MOY). For each dimension, the eta-squared (η^2) effect size was computed as the proportion of the total variance in call volume explained by the seasonal grouping; it ranges from 0 (the grouping explains none of the variation) to 1 (it explains all of it). Using the common effect-size thresholds, a dimension was classified as significant when $\eta^2 \geq 0.06$, corresponding to at least a medium effect.

The results reveal a strongly dominant intraweek seasonal pattern. Of the 175 retained skills, 174 skills (99.4%) exhibit significant day-of-week seasonality. The DOW effect size distribution further shows that this weekly pattern is not only widespread but also strong: 155 skills have a large DOW effect ($\eta^2 \geq 0.14$), 19 have a medium effect ($0.06 \leq \eta^2 < 0.14$), and only 1 skill has a small effect ($0.01 \leq \eta^2 < 0.06$). Month-of-year seasonality is present for 58 skills (33.1%), while day-of-month seasonality is not significant for any skill. The MOY effect size distribution is weaker than the DOW distribution, with 113 skills classified as small, 48 as medium, 10 as large, and 4 as negligible ($\eta^2 < 0.01$).

Based on the significance threshold, skills were segmented into three groups. Segment A contains 57 skills (32.6%) with both significant weekly and annual seasonality. Segment B contains 117 skills (66.9%) with weekly seasonality only. Segment C contains 1 skill (0.6%) with annual seasonality only. No skills were classified as having significant intra-month seasonality. The overwhelming dominance of DOW seasonality confirms that intraweek demand variation is the primary structural driver across the portfolio and justifies the use of a weekly seasonal period in subsequent forecasting models.

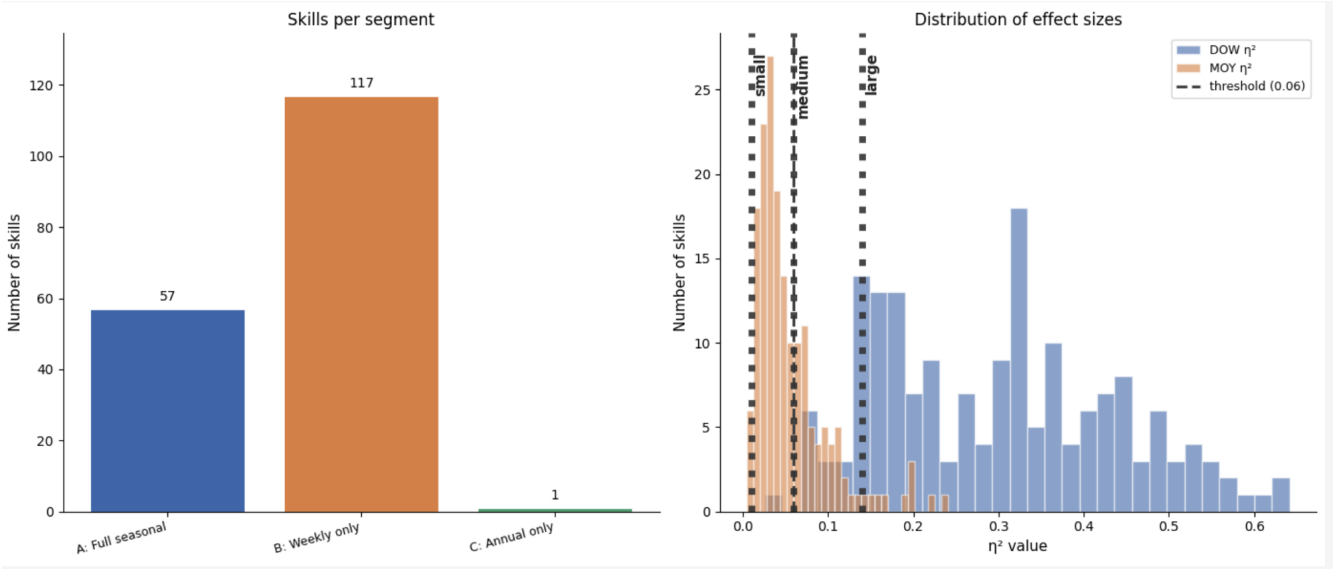


Figure 3.11: Seasonality segmentation results based on η^2 effect sizes for 175 retained skills. The bar chart on the left shows the number of skills assigned to each segment: full seasonal, weekly only, and annual only. The histogram on the right shows the distribution of DOW and MOY η^2 values, with the dashed vertical line indicating the significance threshold of 0.06.

3.3.4 Assessment of Additive vs. Multiplicative Structure

Before applying time series similarity measures and forecasting models, it is necessary to determine whether each skill time series follows an additive or multiplicative structure. This distinction is important because the appropriate error and seasonality specification affects model fit and forecast accuracy. In an additive structure, fluctuations around the trend are approximately constant in magnitude regardless of the level of the series. In a multiplicative structure, fluctuations scale proportionally with the level, meaning that variability increases when demand is high and decreases when demand is low.

To identify the appropriate structure for each skill, a model-based selection procedure was implemented using Exponential Smoothing (ETS). The final 30% of the time series was preserved as an untouched test period, starting on 19 August 2025, while model selection was performed only within the first 70% of the data. Within this model-selection window, candidate models were evaluated on a validation segment using the Sum of Absolute Errors (SAE). The validation score was computed using both real and imputed validation values, while the final test period remained unused during model selection.

Two ETS specifications were compared for each skill. The additive specification was fitted on the original scale using a linear additive structure. The multiplicative specification was implemented through a logarithmic transformation: the series was transformed using $\log(1+x)$, an ETS model was fitted on the transformed scale, and forecasts were back-transformed using $e^x - 1$. Both specifications used a weekly seasonal period of seven days to capture the dominant intraweek demand pattern identified in the seasonality analysis. The model with the lower validation SAE was selected as the preferred specification for each skill.

Under this framework, 108 of the 175 retained skill series (61.7%) were classified as multiplicative-log, while 67 skills (38.3%) were classified as additive-linear, as shown in Figure 3.12. The

scatter plot of per-skill additive SAE versus multiplicative-log SAE shows that many skills lie close to the diagonal, indicating that the two specifications often perform similarly. However, the multiplicative-log model wins for a majority of skills. The distribution of SAE differences, defined as additive SAE minus multiplicative-log SAE, has a median value of 12.96, meaning that the multiplicative-log specification provides a modest overall advantage. The boxplots show broadly comparable SAE distributions across the two model types, with substantial variation across skills.

These results indicate that both additive and multiplicative structures are present in the skill portfolio, but the multiplicative-log specification is more frequently preferred. This finding is consistent with the strong heterogeneity and overdispersion observed in the skill-level demand distributions. Consequently, the multiplicative-log specification is a suitable default modelling choice, while the skill-level model selection results also show that a substantial minority of skills may benefit from an additive-linear specification.

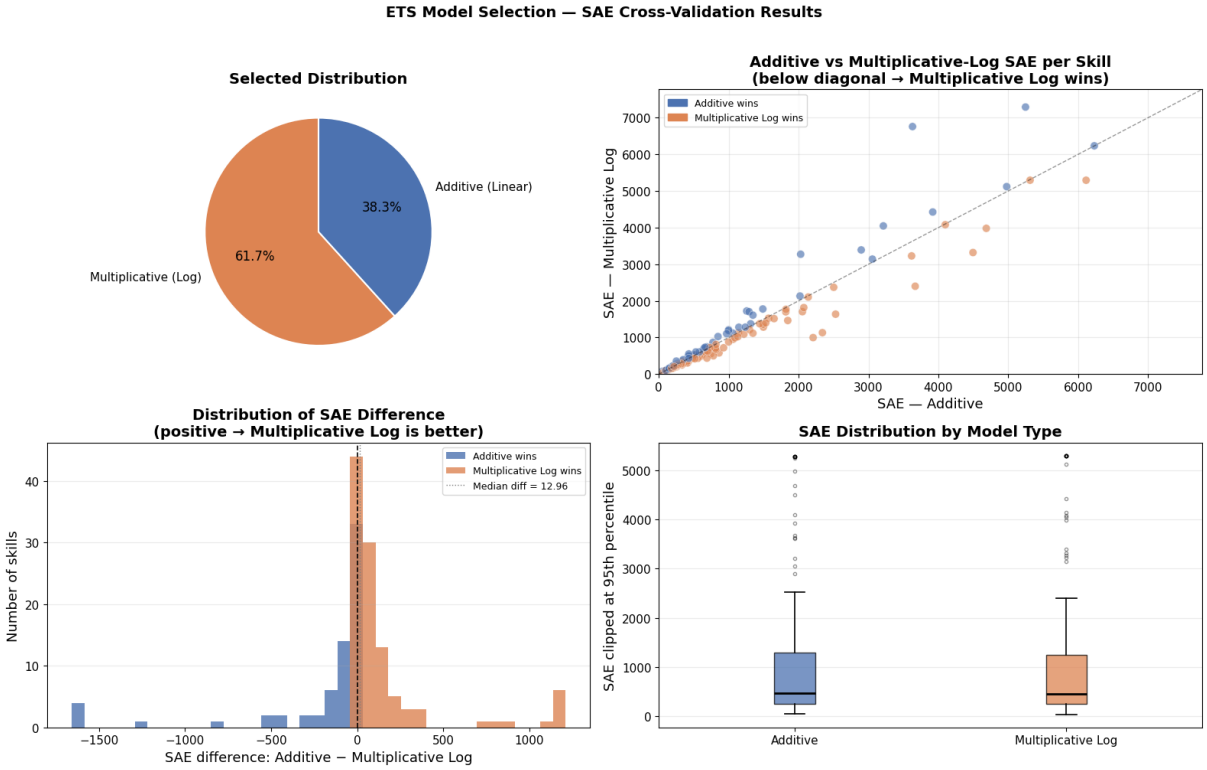


Figure 3.12: ETS model selection results based on SAE validation for 175 retained skills. The pie chart shows the proportion of skills classified as additive-linear or multiplicative-log. The scatter plot compares per-skill additive SAE and multiplicative-log SAE, with points below the diagonal indicating multiplicative-log wins. The histogram shows the distribution of SAE differences, where positive values indicate that the multiplicative-log model performs better. The boxplots summarise SAE distributions by model type across all skills.

Chapter 4

Methodology

Figure 4.1 provides an overview of the methodology followed in this thesis. The colours group the steps by role: grey marks the raw input, teal the data-processing and evaluation stages, purple the forecasting steps, amber the clustering stage, and coral the disaggregation and reconciliation stage.

The pipeline starts from raw call arrival data. In the data cleaning and preprocessing stage, empty, dead, truly new, and sparse skills are removed, and the remaining missing values are imputed, either as zero for skills that are structurally inactive on weekends or using a per-skill ETS model otherwise. Individual skill-level forecasts are then produced with ETS and used as the baseline for evaluation. Next, skills are grouped into clusters: pairwise similarity is measured using four time series similarity measures (DCCA, Scale-Space, DTW, and MSTL) together with a feature-based distance matrix, and three clustering algorithms (k-medoids, hierarchical, and spectral) are applied to these inputs. Forecasts are then produced at the cluster level using ETS and subsequently disaggregated back to the individual skill level using two methods: top-down disaggregation based on day-of-week proportions, and MinT reconciliation. Finally, the resulting skill-level forecasts are evaluated using the Sum of Absolute Errors (SAE) and compared against the baseline.

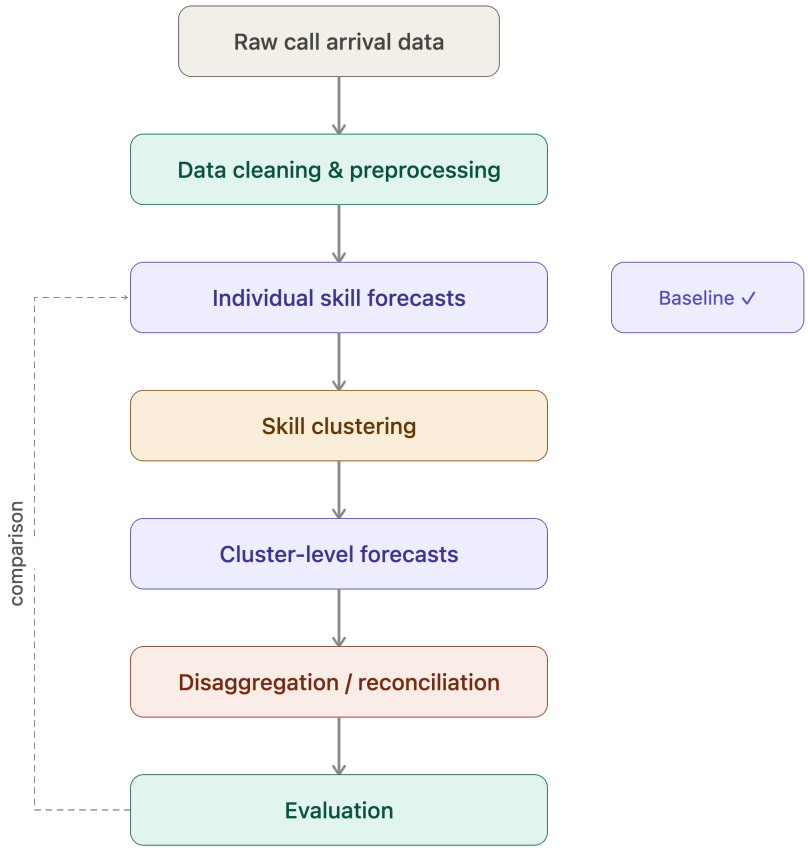


Figure 4.1: Overview of the proposed methodology. The pipeline moves from raw data through preprocessing, individual-skill baseline forecasting, data-driven clustering, cluster-level forecasting, and reconciliation, and then evaluation.

4.1 Forecasting Method

This section describes the forecasting method that underpins both the individual-skill baseline and the cluster-level forecasts. It begins with the general exponential smoothing (ETS) framework, then specifies the particular model structure chosen for this study, and concludes by defining the evaluation metric used to compare forecasts.

4.1.1 Exponential Smoothing

Exponential smoothing is a family of time series forecasting methods that generate forecasts as weighted averages of past observations, where the weights decay exponentially as observations recede further into the past. Unlike simple moving average methods that assign equal

weight to all observations within a fixed window, exponential smoothing places greater emphasis on recent observations while still retaining information from the full history of the series. This property makes exponential smoothing particularly well suited to operational demand forecasting, where recent demand levels are generally more informative about future behaviour than distant historical observations. The general framework for exponential smoothing is the Error-Trend-Seasonality (ETS) model [9], which decomposes the forecasting problem into three structural components: the error term, the trend, and the seasonal pattern. Each component can be specified as either additive or multiplicative, or absent, leading to a family of models that can be tailored to the structural characteristics of the series being forecast. In the additive error formulation, the observation equation is

$$x_t = \hat{x}_t + e_t, \tag{4.1}$$

where $\hat{x}_t$ is the one-step-ahead forecast of x_t given information up to time $t-1$, and e_t is an additive error term. In the multiplicative error formulation,

$$x_t = \hat{x}_t(1 + e_t), \tag{4.2}$$

so that the magnitude of the error scales with the level of the series. The Holt-Winters seasonal extension of exponential smoothing [7] explicitly models both trend and seasonal components through three updating equations. Let ℓ_t denote the level, b_t the trend, and s_t the seasonal component at time t, and let m denote the seasonal period. In the additive formulation, the updating equations are

$$\ell_t = \alpha(x_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}), \tag{4.3}$$

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}, \tag{4.4}$$

$$s_t = \gamma(x_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \tag{4.5}$$

where $\alpha \in (0,1)$, $\beta \in (0,1)$, and $\gamma \in (0,1)$ are smoothing parameters controlling the rate of adaptation of the level, trend, and seasonal components respectively. The recursions are initialised by the level ℓ_0, the trend b_0, and the initial seasonal components $s_{-(m-1)}, \ldots, s_0$, which together with the smoothing parameters α, β, and γ are estimated by maximum likelihood from the training data. The h-step-ahead forecast is then

$$\hat{x}_{t+h} = \ell_t + hb_t + s_{t+h-m}. \tag{4.6}$$

4.1.2 Model Specification

Based on the findings of the exploratory data analysis presented in the previous chapter, two structural decisions are made for the forecasting models applied in this study.

First, a weekly seasonal period of $m = 7$ is used for all skills. As established in the seasonality analysis (Section 3.3.3), day-of-week variation is the dominant and most consistent seasonal driver across the skill portfolio, with 174 of the 175 skills (99.4%) exhibiting significant and predominantly strong day-of-week seasonality, while intra-month seasonality is absent and

intra-year seasonality is present for only a minority of skills. Incorporating a weekly seasonal period directly into the model structure therefore ensures that forecasts respect the systematic differences in demand levels across days of the week.

Second, the forecasting specification is determined by a cross-validation-based selection between an additive-linear and a multiplicative-log structure. As shown in Section 3.3.4, both structures are present in the portfolio, with the multiplicative-log specification preferred for the majority of skills (61.7%) and the additive-linear specification for the remainder. To assess the impact of this choice at the portfolio level, the total Sum of Absolute Errors (SAE) was computed for three strategies: a uniform additive-linear specification, a uniform multiplicative-log specification, and a cross-validation-selected (CV-selected) strategy that adopts the better of the two for each skill individually. The CV-selected strategy achieves the lowest total error (SAE 775,126), compared with 781,936 for the uniform multiplicative-log and 818,296 for the uniform additive-linear specification. Because the per-skill choice can never perform worse than the better uniform strategy, the CV-selected approach is adopted. The improvement over the uniform multiplicative-log specification is modest (roughly 0.9%), reflecting that multiplicative-log is the appropriate structure for most skills, while the larger gap over the additive-linear specification (around 5%) shows the cost of forcing an additive structure on skills whose variability scales with demand. For skills assigned the multiplicative-log specification, the multiplicative structure is handled practically by applying a $\log(1 + x)$ transformation before fitting an additive ETS model, and back-transforming forecasts using $e^x - 1$. This stabilizes variance, prevents negative forecasts, and handles near-zero demand values gracefully.

Formally, for a skill series $\{x_t\}_{t=1}^{N}$ assigned the multiplicative-log specification, the transformed series is defined as

$$x_t^* = \log(1 + x_t), \tag{4.7}$$

and an ETS model with additive error, additive trend, and additive weekly seasonality, ETS(A,A,A), is fitted on $\{x_t^*\}$ with seasonal_periods = 7. The h-step-ahead forecast on the original scale is recovered as

$$\hat{x}_{t+h} = \exp(\hat{x}_{t+h}^*) - 1, \tag{4.8}$$

and all forecasts are clipped to be non-negative. Skills assigned the additive-linear specification are fitted directly on the original scale using the same ETS(A,A,A) structure and weekly seasonal period.

4.1.3 Forecast Evaluation

Model performance is evaluated using the Sum of Absolute Errors (SAE), defined as

$$\text{SAE} = \sum_{t \in \mathcal{T}_{\text{test}}} |x_t - \hat{x}_t|, \tag{4.9}$$

where $\mathcal{T}_{\text{test}}$ is the set of test-period time indices, x_t is the observed call volume, and $\hat{x}_t$ is the corresponding forecast. For each skill, the smoothing parameters and initial states of the ETS model are estimated once on the training period (the first 70% of observations), and the fitted model is then used to forecast the entire test horizon in a single multi-step forecast. The forecast

for a test day at horizon h is therefore $\hat{x}_{T+h}$, made from the final training day T, and no test-period observations are used to re-estimate or update the model. As a result, each $\hat{x}_t$ is a genuine out-of-sample forecast, and forecasts for later test days correspond to longer horizons.

SAE was selected because it measures forecast error directly on the original call-volume scale, making it easy to interpret operationally as the total number of calls misforecast over the evaluation period. It also avoids the instability of percentage-based metrics such as MAPE, which can become undefined or misleading when actual demand is zero or close to zero. This is important for low-volume skills and weekend observations, where near-zero demand is common.

Although SAE is scale-dependent, this is appropriate for the comparisons in this study because models are evaluated on the same skill and the same test period. At the portfolio level, SAE naturally gives more weight to high-volume skills, where forecast errors have greater operational impact for staffing and service-level planning.

Evaluation is performed exclusively on originally observed values using the `is_real` flag introduced during preprocessing. This ensures that imputed values do not contribute to the reported forecast error.

4.2 Similarity Measures

In order to cluster contact center skills based on their historical call arrival behaviour, it is necessary to define appropriate similarity (or distance) measures between time series. The dataset consists of daily call volumes per skill over roughly two years. A time series can differ from another in several distinct ways. Two skills may differ in their overall volume level, in their long-term trend, in the strength and shape of their seasonal cycles, in the shape of their demand curve over time, or in when and how often they show extreme spikes. No single measure captures all of these aspects equally well.

Outliers were not removed prior to similarity computation. The rationale is that if two skills exhibit extreme deviations at the same points in time, this behaviour itself is informative and may indicate shared operational drivers or external factors. Therefore, preserving outliers allows the similarity measures to capture correlated anomalies rather than artificially smoothing them away.

A naive approach would be to compute the Pearson correlation on the raw daily volumes. However, this approach has several limitations. A single large spike can dominate the metric, two series with similar seasonal patterns but different scales may appear weakly correlated, and long-term trends can obscure weekly or monthly structures. As discussed in [23], standard Pearson correlation often mixes or overlooks these differences in non-stationary time series.

For this reason, multiple complementary measures are considered, each capturing a different aspect of the series: Detrended Cross-Correlation Analysis (long-range co-movement, Section 4.2.1), Scale-Space Multiresolution Correlation Analysis (correlation across scales, Section 4.2.2), Dynamic Time Warping (demand shape under time alignment, Section 4.2.3), MSTL-based component similarity (Section 4.2.4), and a feature-based distance (operational profile, Section 4.2.5).

4.2.1 Detrended Cross-Correlation Analysis (DCCA)

Detrended Cross-Correlation Analysis (DCCA) is a statistical technique developed to measure the relationship between two time series that may be non-stationary, meaning their statistical properties such as mean and variance change over time [16, 23, 31, 20]. Traditional correlation measures, such as Pearson correlation, assume stationarity and can therefore be misleading when applied to data containing trends, cycles, or gradual structural changes. DCCA addresses this limitation by explicitly removing local trends before quantifying the degree to which two series co-fluctuate. A key strength of DCCA is that it evaluates correlation as a function of time scale, allowing researchers to distinguish between short-term interactions and long-term dependencies within the same analytical framework. This multi-scale capability makes DCCA particularly suitable for complex real-world signals that exhibit persistence, seasonality, or long-range memory rather than remaining stable around a fixed average.

In operational environments such as contact center management, daily call volumes across different skills or queues often evolve over time due to external drivers like marketing campaigns, business growth, product launches, or seasonal demand cycles. DCCA filters out these common trends and focuses instead on synchronized fluctuations around local baselines. Consequently, it enables a more reliable identification of genuine inter-skill dependencies and helps decision-makers understand whether connections occur primarily in the short term, the long term, or across multiple horizons. Such information is valuable for forecasting, workforce planning, and cross-training strategies where understanding temporal structure is as important as measuring overall similarity.

Consider two discrete time series $\{x_t\}_{t=1}^{N}$ and $\{y_t\}_{t=1}^{N}$ of equal length N, with sample means $\bar{x}$ and $\bar{y}$. DCCA begins by forming the cumulative (integrated) profile of each mean-subtracted series,

$$X_k = \sum_{t=1}^{k}(x_t - \bar{x}), \qquad Y_k = \sum_{t=1}^{k}(y_t - \bar{y}), \qquad k = 1, 2, \ldots, N. \tag{4.10}$$

Subtracting the mean ensures that the accumulated values represent deviations rather than absolute levels, while the cumulative summation (referred to as integration, by analogy with the continuous case) transforms many forms of non-stationarity into smoother components that can be locally approximated and removed. To examine dependence at a particular time scale s, the profiles are divided into overlapping windows of length s. Within each window, a least-squares polynomial fit is used to estimate local trends, denoted $\tilde{X}_{k,s}$ and $\tilde{Y}_{k,s}$, which are then subtracted to obtain detrended residuals. The local detrended covariance for a window beginning at index i is defined as

$$f^2_{\mathrm{DCCA}}(s, i) = \frac{1}{s+1}\sum_{k=i}^{i+s}(X_k - \tilde{X}_{k,s})(Y_k - \tilde{Y}_{k,s}). \tag{4.11}$$

Averaging this quantity across all overlapping windows yields the global detrended covariance function,

$$F^2_{\mathrm{DCCA}}(s) = \frac{1}{N-s}\sum_{i=1}^{N-s} f^2_{\mathrm{DCCA}}(s, i). \tag{4.12}$$

The quantity $F^2_{\mathrm{DCCA}}(s)$ cannot be used on its own, because it depends on the size of each series and is not comparable across skill pairs. To make it bounded and comparable, it is

divided by the fluctuation of each series on its own, taken from Detrended Fluctuation Analysis (DFA). DFA is the single-series version of DCCA: instead of the joint fluctuation of two series, it measures how much one series fluctuates around its own local trend. The DFA fluctuation $F_{\mathrm{DFA},x}(s)$ is computed in the same way as before, but using the squared term $(X_k - \tilde{X}_{k,s})^2$ instead of the cross term, and gives the typical size of the detrended fluctuations of $\{x_t\}$ at scale s (and the same for $\{y_t\}$). The DCCA cross-correlation coefficient is then

$$\rho_{\mathrm{DCCA}}(s) = \frac{F^2_{\mathrm{DCCA}}(s)}{F_{\mathrm{DFA},x}(s)\, F_{\mathrm{DFA},y}(s)}, \qquad -1 \le \rho_{\mathrm{DCCA}}(s) \le 1, \tag{4.13}$$

which behaves like an ordinary correlation but is scale-dependent and robust to trends [16, 23, 31].

In this thesis, $\rho_{\mathrm{DCCA}}(s)$ is computed directly from the detrended cross-covariance and the single-series detrended fluctuations, as in the formula above. The call volumes are first $\log(1+x)$-transformed, a linear local detrending is applied, and the coefficient is computed at four scales, with windows of $s \in \{7, 14, 30, 365\}$ days, spanning weekly to annual horizons. The final DCCA similarity between two skills is the average of $\rho_{\mathrm{DCCA}}(s)$ over these four scales, giving one value per skill pair, and the corresponding distance is $1 - \rho_{\mathrm{DCCA}}$.

4.2.2 Scale-Space Multiresolution Correlation Analysis

Scale-Space Multiresolution Correlation Analysis is a time series methodology designed to uncover correlation structures that vary both over time and across temporal scales. Unlike classical correlation, which produces a single global coefficient summarizing average linear dependence over the full observation period, this approach assumes that two signals may be positively correlated at some horizons, negatively correlated at others, and possibly unrelated during certain intervals. The method therefore combines three key ideas: multiresolution decomposition, local (time-varying) correlation, and statistical credibility assessment. In practice, it is particularly suitable for non-stationary operational data, such as daily contact center volumes across skills, where trends, seasonality, and short-term fluctuations coexist and where relationships between series are not constant through time [22]. The mathematical formulation below follows the framework introduced by Pasanen and Holmström.

The method proceeds in three stages. First, each series is decomposed into components at different temporal scales, so that so that short-term detail and long-term structure can be examined separately. Second, for each scale, the local correlation between the two series is computed at every point in time, giving a time-by-scale map of co-movement. Third, Bayesian inference is used to assess which parts of this map are statistically trustworthy rather than the result of noise, since with noisy, autocorrelated demand data two skills can appear locally correlated purely by chance.

The first stage is a multiresolution decomposition of each time series into additive components representing distinct temporal scales. Let $\{x_t\}_{t=1}^N$ and $\{y_t\}_{t=1}^N$ be two series of equal length N observed at times $t_1 < \cdots < t_N$. Define a smoothing operator S_λ indexed by a non-negative

smoothing parameter λ. In the spline-based formulation,

$$S_\lambda = (I + \lambda C^T C)^{-1}, \tag{4.14}$$

where I is the identity matrix and C is a second-order differencing matrix that penalizes curvature. As λ increases, the signal becomes progressively smoother and converges toward a linear trend. Choosing a sequence of smoothing levels

$$0 = \lambda_1 < \lambda_2 < \cdots < \lambda_L \leq \infty, \tag{4.15}$$

the original series can be expressed as a sum of scale-dependent components,

$$\mathbf{x} = \sum_{j=1}^{L} u_j, \qquad u_j = (S_{\lambda_j} - S_{\lambda_{j+1}})\mathbf{x} \;\; (j < L), \quad u_L = S_{\lambda_L}\mathbf{x}, \tag{4.16}$$

and analogously $\mathbf{y} = \sum_{j=1}^{L} v_j$, where $\mathbf{x} = (x_1, \ldots, x_N)^T$ and $\mathbf{y} = (y_1, \ldots, y_N)^T$. This identity is not circular: each u_j is computed directly from the known series $\mathbf{x}$, and the equation simply reconstructs $\mathbf{x}$ from these pieces. The idea is to take progressively smoother versions of the series (with $\lambda_1 = 0$, so $S_{\lambda_1}\mathbf{x} = \mathbf{x}$), where each u_j is the difference between two consecutive smoothing levels and captures the structure lost at that step, and u_L is the smoothest part that remains. The sum telescopes, so the components add back exactly to $\mathbf{x}$. Because u_j and v_j are taken at the same smoothing level λ_j, they represent the same temporal scale in the two series, from short-term noise to long-term trend, and can be compared directly. This prevents large-scale behavior from masking fine-scale relationships, which matters when demand data show both daily variability and long-term growth.

The second stage evaluates local correlation within each scale component. Instead of computing a single Pearson coefficient, correlation is estimated at each time point using a weighted window centered at that time. Let $w_k = [w_{1k}, \ldots, w_{Nk}]^T$ be normalized weights satisfying $\sum_{t=1}^{N} w_{tk} = 1$. Here k indexes the time point t_k at which the correlation is evaluated: the weights w_k are centered on t_k, so that observations near t_k receive most weight and distant observations little. The quantities below are therefore local estimates around time t_k rather than global summaries, and repeating the calculation for every $k = 1, \ldots, N$ produces a separate correlation value at each point in time, which is what makes the analysis time-varying. The local weighted mean and variance **of** $\{x_t\}$ at time t_k are

$$E_W(x; k) = w_k^T \mathbf{x}, \qquad \mathrm{Var}_W(x; k) = \sum_{t=1}^{N} w_{tk}\big(x_t - E_W(x; k)\big)^2, \tag{4.17}$$

and the local covariance and correlation between $\{x_t\}$ and $\{y_t\}$ are

$$\mathrm{Cov}_W(x, y; k) = \sum_{t=1}^{N} w_{tk}\big(x_t - E_W(x; k)\big)\big(y_t - E_W(y; k)\big), \tag{4.18}$$

$$\mathrm{Corr}_W(x, y; k) = \frac{\mathrm{Cov}_W(x, y; k)}{\sqrt{\mathrm{Var}_W(x; k)}\sqrt{\mathrm{Var}_W(y; k)}}. \tag{4.19}$$

The weights are typically derived from a biweight kernel with scale parameter σ, which controls the temporal window width. Small σ captures short-term co-movements, while large σ reveals long-term associations. Repeating this calculation for multiple σ values yields a two-dimensional correlation map with time on one axis and scale on the other, allowing simultaneous visualization of intra-week, intra-month, and long-horizon dependencies in call volume data.

The final stage introduces statistical credibility through Bayesian inference. Its purpose is to filter the time-by-scale correlation map from the second stage, keeping only the regions that are unlikely to be due to chance. The observed series are modeled as noisy realizations,

$$\mathbf{x} = \boldsymbol{\mu} + \boldsymbol{\varepsilon}, \qquad \mathbf{y} = \boldsymbol{\nu} + \boldsymbol{\delta}, \tag{4.20}$$

where $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^N$ are latent smooth signal vectors and $\boldsymbol{\varepsilon}, \boldsymbol{\delta} \in \mathbb{R}^N$ are stochastic error vectors. In other words, each observed series is seen as an unknown true signal ($\boldsymbol{\mu}$ or $\boldsymbol{\nu}$) plus random noise. Instead of computing the correlation map only once, the method draws many plausible versions of the smooth signals from their posterior distribution and recomputes the components and their local correlations for each one. This gives a distribution of correlation values at each time point and scale, rather than a single number. The method then keeps the time-scale regions where the correlation stays clearly positive or negative across these draws, with high posterior probability. This reduces false findings caused by noise or autocorrelation and produces credibility maps that separate reliable relationships from random ones.

For contact center demand analysis, this technique is highly relevant because skill queues often share seasonal structures while diverging in short-term fluctuations due to campaigns, outages, or policy changes. Traditional correlation cannot separate these effects, whereas scale-space multiresolution correlation reveals whether two skills are aligned only during peak seasons, only during daily spikes, or consistently over long horizons. The method therefore supports more informed forecasting, staffing alignment, and cross-training strategies by exposing the full temporal architecture of inter-skill dependence rather than compressing it into a single summary statistic [22].

4.2.3 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) [14] is a distance-based similarity measure designed to compare two time series by allowing flexible, non-linear alignments along the time axis. Instead of forcing observations at identical time indices to be matched, DTW permits local stretching and compression of segments so that similar shapes can be aligned even if they occur at slightly different times. This property makes DTW especially useful for operational demand data such as call-center queues, where peaks, drops, or campaign effects may appear a few days earlier or later between skills while still representing the same underlying behavioural pattern. In contrast to correlation-based methods that emphasize linear co-movement or synchronized trends, DTW focuses primarily on geometric shape similarity and is therefore robust to temporal offsets and unequal sequence lengths.

Consider two time series $\{x_t\}_{t=1}^{N}$ and $\{y_t\}_{t=1}^{M}$, with possibly different lengths N and M. DTW constructs an $N \times M$ cost matrix in which each element represents the pointwise distance

between samples,

$$d_{t,j} = \delta(x_t, y_j), \tag{4.21}$$

where the index $t \in \{1, \ldots, N\}$ refers to a time point in the first series $\{x_t\}$ and the index $j \in \{1, \ldots, M\}$ refers to a time point in the second series $\{y_t\}$, so that $d_{t,j}$ is the distance between the t-th observation of the first series and the j-th observation of the second. The function $\delta(\cdot, \cdot)$ is typically the absolute or squared Euclidean distance. The objective is to find a warping path

$$W = \{(t_1, j_1), (t_2, j_2), \ldots, (t_L, j_L)\}, \tag{4.22}$$

that aligns the two sequences while minimizing the cumulative alignment cost. The path must satisfy boundary conditions $(1, 1)$ to (N, M), monotonicity, and continuity constraints so that time ordering is preserved. Using dynamic programming, the cumulative cost matrix C is computed recursively. As in the cost matrix d, the row index $t \in \{1, \ldots, N\}$ corresponds to a time point of the first series $\{x_t\}$ and the column index $j \in \{1, \ldots, M\}$ to a time point of the second series $\{y_t\}$, so that each entry $C_{t,j}$ is the smallest possible total cost of aligning the first t observations of $\{x_t\}$ with the first j observations of $\{y_t\}$. The recursion is

$$C_{t,j} = d_{t,j} + \min\{C_{t-1,j},\ C_{t,j-1},\ C_{t-1,j-1}\}, \tag{4.23}$$

which states that the best alignment ending at the pair (x_t, y_j) adds the local distance $d_{t,j}$ to the cheapest of the three preceding alignments: moving one step in the first series ($C_{t-1,j}$), one step in the second series ($C_{t,j-1}$), or one step in both ($C_{t-1,j-1}$). The recursion is initialized along the first row and column, with

$$C_{1,1} = d_{1,1}, \qquad C_{t,1} = d_{t,1} + C_{t-1,1}, \qquad C_{1,j} = d_{1,j} + C_{1,j-1}, \tag{4.24}$$

since the first row and column allow movement in only one series. The DTW distance between $\{x_t\}$ and $\{y_t\}$ is then the value in the final cell, $C_{N,M}$, which represents the minimal total cost of aligning the two complete sequences under the allowed time distortions. This formulation ensures that local time shifts are absorbed into the alignment path rather than being treated as mismatches.

Conceptually, DTW can be interpreted as sliding and bending one curve along the time axis until it best overlaps with the other. In general, this is useful when two series have the same demand shape but their peaks occur at slightly different times, since a standard Euclidean or Pearson measure would penalize such shifts heavily, whereas DTW recognizes the series as structurally similar. DTW also supports comparisons of unequal-length sequences, which is helpful when series have missing periods or different observation horizons.

In this study, DTW helps in some cases but not others. The regular weekly cycle is calendar-locked, with almost all skills peaking on the same weekdays (Section 3.3.3), and the series are aligned to a common daily calendar and equal length after preprocessing, so for the dominant weekly pattern there is little timing misalignment for DTW to correct. However, demand is also driven by irregular events such as marketing campaigns, invoicing or billing cycles, product launches, and outages, which do not follow the weekly calendar and can affect related skills a

few days apart rather than on exactly the same day. For these shifted but related spikes, DTW's flexible alignment can recognize skills as similar where a strictly calendar-aligned measure would not. DTW is therefore included to capture this kind of shape-based, time-shifted co-movement, even though its advantage for the regular weekly pattern is limited.

The main trade-off is computational complexity, since the dynamic programming matrix grows with NM, but for moderate daily series lengths this cost remains manageable, and the alignment-based shape comparison can be more informative than a single global correlation coefficient.

4.2.4 Multiple Seasonal-Trend Decomposition using Loess (MSTL)

MSTL [2] separates a series $\{x_t\}_{t=1}^{N}$ into interpretable structural components before similarity is computed. In this thesis a single weekly seasonal cycle is used, consistent with the dominant day-of-week pattern identified in Section 3.3.3, so the decomposition is additive with one seasonal component,

$$x_t = \mathcal{T}_t + \mathcal{S}_t + \mathcal{R}_t, \tag{4.25}$$

where $\mathcal{T}_t$ is the trend, $\mathcal{S}_t$ the weekly seasonal component, and the remainder is $\mathcal{R}_t = x_t - \mathcal{T}_t - \mathcal{S}_t$. The additive or multiplicative structure of each skill is determined once, using the ETS-based identification procedure of Section 3.3.4; skills identified as multiplicative are log-transformed before decomposition, so that MSTL is always applied to an additive structure.

After decomposition, the similarity between two series $\{x_t\}_{t=1}^{N}$ and $\{y_t\}_{t=1}^{N}$ is computed separately for each component using Pearson correlation,

$$\rho_{\mathcal{T}} = \mathrm{Corr}(\mathcal{T}_t^x, \mathcal{T}_t^y), \quad \rho_{\mathcal{S}} = \mathrm{Corr}(\mathcal{S}_t^x, \mathcal{S}_t^y), \quad \rho_{\mathcal{R}} = \mathrm{Corr}(\mathcal{R}_t^x, \mathcal{R}_t^y), \tag{4.26}$$

where superscripts x and y denote components extracted from $\{x_t\}$ and $\{y_t\}$ respectively. The final similarity score is the average of the three component correlations,

$$\rho_{\mathrm{MSTL}}(x, y) = \frac{1}{3}\left(\rho_{\mathcal{T}} + \rho_{\mathcal{S}} + \rho_{\mathcal{R}}\right). \tag{4.27}$$

This component-wise comparison reduces the influence of scale differences, random spikes, and long-term drift, so that skills are grouped according to persistent and interpretable temporal patterns rather than raw magnitude or transient anomalies.

4.2.5 Feature-Based Distance Measure

Whereas the time series similarity measures described in the preceding sections operate directly on the temporal structure of the raw call volume series, the feature-based distance measure represents each skill by a compact vector of forecast-relevant statistics extracted from its training window, and supplements this with two pairwise measures of co-occurrence in idle and exceptional days. This representation captures characteristics that are operationally meaningful for forecasting but may not be apparent from the shape of the raw series alone, such as relative volatility, linear trend, weekend inactivity, the additive or multiplicative character of the series, and the shape of the weekly demand profile.

Feature Extraction: For each skill series $\{x_t\}_{t=1}^{N}$, let $\{x_t\}_{t=1}^{N_{\text{train}}}$ denote the training window comprising the first 70% of daily observations, where missing days within a skill's active range are treated as zero demand. Skills with fewer than 21 training days are excluded from feature extraction. The following features are computed.

Coefficient of variation measures relative demand volatility:

$$\text{cv} = \frac{\sigma_{\text{train}}}{\max(\bar{x}_{\text{train}},\, 1)}, \tag{4.28}$$

where σ_{train} and $\bar{x}_{\text{train}}$ are the standard deviation and mean of the training series. The denominator is floored at one to avoid inflating the ratio for very low-volume skills.

Normalised trend slope captures the direction and magnitude of the linear trend relative to the mean volume:

$$\text{trend\_slope} = \frac{\hat{\beta}}{\max(\bar{x}_{\text{train}},\, 1)}, \tag{4.29}$$

where $\hat{\beta}$ is the ordinary least-squares slope of x_t regressed on the time index t, which equals zero for a constant series. Normalising by the (floored) mean expresses trend relative to demand level rather than in absolute terms.

Weekend zero fraction measures the degree to which a skill is inactive at weekends:

$$\text{weekend\_zero\_pct} = \frac{1}{|W|} \sum_{t \in W} \mathbf{1}[x_t = 0], \tag{4.30}$$

where W is the set of Saturdays and Sundays in the training window.

Sunday-only zero asymmetry distinguishes six-day from five-day operating patterns. Let p_{Sat} and p_{Sun} denote the fractions of zero-demand on Saturdays and Sundays respectively. Then

$$\text{sunday\_only\_zero} = \max(p_{\text{Sun}} - p_{\text{Sat}},\, 0)\,, \tag{4.31}$$

which is large for skills that are systematically closed on Sundays but still operate on Saturdays.

Level–volatility correlation and multiplicative indicator characterise whether variability scales with the level of the series. Let μ_t and s_t be the seven-day rolling mean and rolling standard deviation. A seven-day window is used so that each rolling statistic spans exactly one full week and therefore includes every day of the week once, this removes the strong day-of-week pattern from the rolling mean and standard deviation, so that changes in μ_t and s_t reflect genuine shifts in level and volatility rather than the weekly cycle. A window that is not a multiple of seven would mix unequal numbers of weekdays and weekend days and introduce spurious fluctuations driven purely by seasonality. The shorter window does make each statistic noisier, but this is averaged out over the full training period when the correlation below is computed. The level–volatility correlation is

$$\text{level\_vol\_corr} = \rho = \text{corr}(\mu_t,\, s_t)\,, \tag{4.32}$$

computed over windows with $\mu_t > 0$, and the binary multiplicative indicator is

$$\text{is\_multiplicative} = \mathbf{1}[\rho \geq 0.35]\,. \tag{4.33}$$

A strong positive correlation indicates multiplicative behaviour, in which variability grows with demand. Both the continuous correlation and the binary indicator are retained as features.

Day-of-week profile is a seven-dimensional vector of normalised mean volumes per day of the week:

$$\mathrm{dow}_d = \frac{\bar{x}_d}{\sum_{d'=0}^{6} \bar{x}_{d'}}, \qquad d = 0, 1, \ldots, 6, \tag{4.34}$$

where $\bar{x}_d$ is the mean volume on day of week d (Monday to Sunday). The profile sums to one and describes how demand is distributed across the days of the week, independently of overall volume, so two skills with the same weekly shape but different sizes have the same profile. When a skill has negligible total volume, and the weekly shape cannot be estimated reliably, a flat profile of $1/7$ on each day is used, corresponding to demand spread evenly across the week.

Feature Weighting and Feature Distance: The thirteen features above (six scalar features and the seven day-of-week profile entries) are standardised to zero mean and unit variance using the training data. Differential weights are then applied after standardisation to control the relative influence of each feature. The seven day-of-week profile features are each scaled by a factor of three to emphasise the shape of weekly demand, which the seasonality analysis identified as the dominant structural driver. The normalised trend slope and the weekend zero fraction are each scaled by a factor of two, reflecting their role in separating skills with differing long-run dynamics and weekend operating patterns. The remaining features, coefficient of variation, Sunday-only zero asymmetry, the multiplicative indicator, and the level, volatility correlation, are retained at unit weight.

The feature distance between two skills i and j is the weighted Euclidean distance between their standardised feature vectors,

$$d_{\mathrm{feat}}(i, j) = \sqrt{\sum_{m=1}^{M} w_m^2 \left(\tilde{f}_{i,m} - \tilde{f}_{j,m} \right)^2}, \tag{4.35}$$

where $\tilde{f}_{i,m}$ is the m-th entry of the standardised feature vector of skill i, w_m is the weight applied to feature m, and M is the total number of features. The matrix of pairwise feature distances $d_{\mathrm{feat}}(i, j)$ is then symmetrised, constrained to be non-negative with a zero diagonal, and rescaled by its maximum off-diagonal entry so that all distances lie in $[0, 1]$.

Co-occurrence Distances: Beyond the per-skill feature vector, two pairwise distances capture whether skills share the same idle and exceptional days. Both are evaluated only over the days on which both skills are observed in their training windows.

The *common zero-day distance* compares the days on which each skill records no demand. Let Z_i and Z_j denote the sets of zero-demand days of skills i and j within their common observation window. The distance is the Jaccard distance between these sets,

$$d_{\mathrm{zero}}(i, j) = \frac{|Z_i \triangle Z_j|}{|Z_i \cup Z_j|}, \tag{4.36}$$

that is, the proportion of idle days on which only one of the two skills is zero. It equals zero when

neither skill has any zero days, and equals one when the two skills share no common observation window.

The *common outlier-day distance* applies the same construction to unusually high-volume days. Outliers are flagged per skill using the modified z-score of Iglewicz and Hoaglin [13], a robust criterion based on the median absolute deviation (MAD): a day is flagged if its robust score $0.6745\,(x_t - \text{median})/\text{MAD}$ exceeds 3.5. Here the constant 0.6745 rescales the MAD so that it estimates the standard deviation under normally distributed data (since $\text{MAD} \approx 0.6745\,\sigma$ in that case), which puts the score on the same scale as an ordinary z-score, and 3.5 is the threshold recommended for this score to flag outliers. When the MAD is degenerate (zero), the criterion falls back to an inter-quartile-range rule and then to a standard-deviation rule, and only positive-volume days may be flagged. Letting O_i and O_j be the outlier days of the two skills over their common window, the distance is

$$d_{\text{out}}(i,j) = \frac{|O_i \triangle O_j|}{|O_i \cup O_j|}. \tag{4.37}$$

Composite Distance. The three components defined above, the per-skill feature distance d_{feat}, the common zero-day distance d_{zero}, and the common outlier-day distance d_{out}, are combined into a single feature-based distance as a weighted average,

$$d(i,j) = \frac{w_{\text{f}}\, d_{\text{feat}}(i,j) + w_{\text{z}}\, d_{\text{zero}}(i,j) + w_{\text{o}}\, d_{\text{out}}(i,j)}{w_{\text{f}} + w_{\text{z}} + w_{\text{o}}}, \tag{4.38}$$

with $w_{\text{f}} = 1.0$, $w_{\text{z}} = 0.3$, and $w_{\text{o}} = 0.1$, so that the per-skill feature shape dominates while the co-occurrence of idle and exceptional days provides secondary refinement. The resulting matrix is again symmetrised, made non-negative with a zero diagonal, and rescaled by its maximum off-diagonal entry so that all distances lie in $[0, 1]$. Skills with insufficient training history (fewer than 21 days) are excluded from the computation; in the full matrix returned for clustering, their distances to all other skills are set to the maximum value of one.

This composite distance $d(i,j)$ is the single, final output of the feature-based measure: it is the only quantity used to compare two skills in this track, and the corresponding $K \times K$ distance matrix is what is passed to the clustering algorithms. Unlike the time-series similarity measures, which are also combined into a consensus matrix, the feature-based distance is kept as a separate clustering input throughout.

4.3 Clustering Methods

Each similarity or distance measure from the previous section yields a $K \times K$ distance matrix $D^{(m)} = (d_{ij}^{(m)})$, where similarities are first converted to distances via $d_{ij} = 1 - \text{sim}_{ij}$ and rescaled to $[0, 1]$. These matrices are the input to the clustering stage, in which groups of skills with similar call arrival behaviour are identified. Three clustering approaches are considered: hierarchical clustering, k-medoids, and spectral clustering. They were selected because they represent different clustering philosophies and are all well suited to settings where the main input is a pairwise distance structure rather than low-dimensional feature vectors, and using several methods makes it possible to assess whether the resulting skill groupings are robust to the choice of

algorithm. Hierarchical clustering and k-medoids use the distance matrix directly, while spectral clustering first converts it into an affinity matrix, as described below.

4.3.1 Hierarchical Clustering

Agglomerative hierarchical clustering is a bottom-up clustering method that constructs clusters by repeatedly merging the two most similar clusters until all observations belong to a single cluster [19]. The procedure starts with each skill as its own cluster. At each step, distances between all existing clusters are evaluated, and the pair with the smallest inter cluster distance is merged. This yields a hierarchy of nested partitions.

A key component of the method is the linkage criterion, which determines how the distance between two clusters is calculated from the pairwise distances between their members. Common choices include single linkage, complete linkage, average linkage, and Ward's method. The resulting hierarchy is usually visualized with a dendrogram [3], which shows both the order of cluster merges and the distance level at which they occur.

A main advantage of hierarchical clustering in this setting is that the number of clusters does not need to be specified in advance. This is useful because the appropriate number of skill groups is not known beforehand and may depend on both the structure of the distance matrix and practical forecasting considerations. By examining the dendrogram, multiple candidate partitions can be compared and natural cut points can be identified.

This makes hierarchical clustering well suited to the objectives of this thesis. It supports exploratory analysis when the underlying grouping structure is unknown, provides an interpretable visual representation, and can be applied directly to any of the pairwise distance matrices considered. For these reasons, agglomerative hierarchical clustering is used here as an important baseline and exploratory tool in the clustering stage of the thesis.

4.3.2 K-Medoids Clustering

K-medoids is a partitioning clustering algorithm that divides the set of skills into a pre specified number K of clusters [26]. The method is closely related to k-means, but instead of representing each cluster by its centroid, it represents each cluster by one of the observed data points, called the medoid. The medoid of a cluster is the observation whose average distance to all other observations in the same cluster is minimal. In this way, cluster centers are always actual skills rather than artificial averages.

Given a pairwise distance matrix, the objective of k-medoids is to find K medoids such that the total within-cluster dissimilarity is minimized. Once an initial set of medoids has been chosen, each skill is assigned to the nearest medoid according to the selected distance matrix. The algorithm then iteratively updates the medoids and cluster assignments until no further improvement in the objective function is obtained. The advantage in this setting is that k-medoids requires only the pairwise distances, and not the underlying coordinates of each skill. Most of the similarity measures in this thesis (such as DTW, DCCA, scale-space, and MSTL) produce only a distance between each pair of skills, without placing the skills in a feature space. The distances themselves are therefore well defined, but there are no coordinates to average, so a

cluster mean (as required by k-means) cannot be formed. K-medoids avoids this by representing each cluster with an actual skill (the medoid) rather than an averaged centroid.

A limitation of k-medoids is that the number of clusters must be specified in advance. Unlike hierarchical clustering, the method does not produce a full clustering hierarchy, but only a single partition for a chosen value of K. This means that model selection is required to determine an appropriate number of clusters. In practice, several candidate values of K can be evaluated and compared using internal validation criteria or forecasting performance in downstream analysis. Although this adds an extra modeling decision, it also makes k-medoids attractive when a fixed number of operationally usable skill groups is desired.

In the context of this thesis, k-medoids serves as one of two primary partitioning alternatives alongside hierarchical clustering. Both produce a partition of the skills into K groups, but reach it differently. Hierarchical clustering is greedy and its merges are permanent: once two skills are placed in the same cluster they stay together, and cutting the dendrogram at a chosen level only selects how many merges to keep, so an early grouping cannot be revised. K-medoids is iterative: it repeatedly assigns every skill to its nearest medoid and updates the medoids until the partition stabilises, so skills can move between clusters and the representatives are refined rather than fixed in advance. K-medoids can therefore adjust an initial partition toward a lower within-cluster dissimilarity for a given K and represents each cluster by an actual skill, which aids interpretation, while hierarchical clustering is more exploratory since the full merge structure can be inspected before fixing K. It was included as a robust, distance-based alternative that works naturally with the pairwise distance matrices constructed in the previous section.

4.3.3 Spectral Clustering

Spectral clustering is a graph based clustering method that uses the pairwise similarity structure between observations to identify groups that may not be well separated by traditional distance based partitioning methods [29]. Instead of clustering the observations directly in the original space, spectral clustering first represents the data as a weighted graph in which each skill is a node and the edge weight between two nodes reflects their similarity.

$$a_{ij} = \exp\left(-\frac{d_{ij}^2}{2\sigma_a^2}\right),$$

where d_{ij} denotes the distance between skills i and j, and $\sigma_a > 0$ is the affinity kernel bandwidth controlling the width of the neighborhood.

Once the affinity matrix has been constructed, spectral clustering computes a graph Laplacian, which captures the connectivity structure of the similarity graph. The eigenvectors of this Laplacian define a low dimensional embedding of the skills in which the cluster structure is often more clearly separated than in the original representation. Standard clustering, typically k-means, is then applied in this embedded space to obtain the final partition. The central idea is that the eigenstructure of the graph can reveal grouping patterns that are less apparent from the original pairwise distances alone.

Spectral clustering was selected because it does not impose strong assumptions on cluster shape and can therefore detect complex or non-convex group structures that may not be well

captured by hierarchical linkage rules or medoid based partitioning. This is relevant in the present application because similarities between contact center skills may arise from a combination of seasonal patterns, shared operational shocks, and common long term trends. As a result, the similarity structure may not form simple compact clusters in the usual geometric sense. By focusing on connectivity in the similarity graph, spectral clustering can provide a useful alternative perspective on such relationships.

Like k-medoids, spectral clustering requires the number of clusters to be specified in advance. It therefore does not provide the same exploratory flexibility as hierarchical clustering. In addition, its performance depends on how the similarity graph is constructed and on the choice of tuning parameters such as the scale parameter in the affinity function. Nevertheless, when a suitable value of K is available, spectral clustering can yield informative partitions, especially when the similarity structure is irregular or multi layered.

In summary, spectral clustering was included because it offers a fundamentally different perspective on the clustering problem. While hierarchical clustering builds a nested sequence of partitions and k-medoids minimizes within cluster dissimilarity around representative skills, spectral clustering exploits the global connectivity structure of the similarity graph. This makes it a valuable third method for assessing whether data driven skill groupings are robust across different clustering paradigms.

4.4 Selection of the Number of Clusters

Both k-medoids and spectral clustering require the number of clusters, denoted by K, to be chosen beforehand, and for hierarchical clustering a K-cluster partition is obtained by cutting the dendrogram at a chosen level. Selecting an appropriate value of K is therefore an important step in the clustering procedure, as too few clusters may combine skills with substantially different demand patterns, while too many clusters may fragment structurally similar skills and reduce the practical usefulness of the aggregation.

In this thesis, the number of clusters is evaluated using three internal clustering validity measures: the Silhouette index, the Dunn index, and the C-index. These criteria assess the quality of a partition based solely on the pairwise distance matrix, without requiring external labels. They are therefore well suited to the present setting, where the true grouping of skills is unknown and clustering is performed on data-driven time series distances.

4.4.1 Silhouette Index

The Silhouette index [**rousseeuw1987Silhouettes**] measures how well each observation fits within its assigned cluster compared with the nearest alternative cluster. For a given skill i, let $a(i)$ denote the average distance from i to all other skills in the same cluster. This quantity measures within cluster dissimilarity. Next, for every other cluster, compute the average distance from i to all skills in that cluster, and let $b(i)$ be the minimum of these values. This represents the dissimilarity between skill i and its nearest neighboring cluster. The Silhouette value of skill i is then defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}.$$

This value lies between -1 and 1. Values close to 1 indicate that the skill is well matched to its own cluster and poorly matched to neighboring clusters. Values close to 0 indicate that the skill lies near a cluster boundary, while negative values suggest possible misclassification.

The overall Silhouette index for a clustering solution is obtained by averaging $s(i)$ over all skills. A larger average Silhouette value indicates better cluster separation and cohesion. In this thesis, candidate values of K are compared, and values yielding higher average Silhouette scores are preferred. The Silhouette index is particularly useful because it provides an intuitive balance between compactness and separation, and it is directly applicable to arbitrary distance matrices.

4.4.2 Dunn Index

The Dunn index evaluates clustering quality by comparing the separation between clusters to the compactness within clusters. The idea is that a good clustering solution should produce clusters that are both internally tight and well separated from one another [5].

Let

$$d_{rs} = \min_{i \in C_r,\, j \in C_s} d(i, j), \qquad r \neq s,$$

denote the distance between clusters C_r and C_s, defined as the minimum distance between any pair of skills belonging to different clusters. Let

$$D_t = \max_{i,j \in C_t,\, i \neq j} d(i, j)$$

denote the diameter of cluster C_t, that is, the largest distance between any two skills in the same cluster. Define

$$d_{\min} = \min_{r \neq s} d_{rs}$$

as the minimum inter cluster distance, and

$$d_{\max} = \max_{t} D_t$$

as the maximum within cluster diameter. The Dunn index is then given by

$$D = \frac{d_{\min}}{d_{\max}}.$$

A single-skill cluster contains no within-cluster pairs, so its diameter is taken to be zero. Such a cluster therefore does not contribute to $d_{\max}$, which is determined by the largest non-degenerate cluster. The index remains well defined as long as at least one cluster contains more than one skill, which holds for all partitions considered here.

The numerator represents the smallest separation between any two clusters, while the denominator represents the largest within cluster spread. Large values of the Dunn index are preferred, since they indicate that even the two closest clusters remain well separated relative to the most dispersed cluster.

The Dunn index is attractive in this application because it explicitly penalizes clustering solutions in which at least one cluster is too wide or in which two clusters are too close to one

another. This makes it useful as a complementary criterion to the Silhouette index. However, the measure can be sensitive to outliers, since a single unusually large within cluster distance may inflate the denominator. Because outliers were intentionally retained in this thesis, the Dunn index is not used in isolation, but together with the other validation measures.

4.4.3 C-index

The C-index is another internal validity measure that evaluates how compact the clusters are relative to the range of all pairwise distances in the dataset. Suppose that a clustering solution contains a total of m within-cluster pairwise distances, and let

$$S = \sum_{t} \sum_{\substack{i,j \in C_t \\ i<j}} d_{ij}$$

be the sum of these distances, where the inner sum runs over all pairs of skills within cluster C_t and the outer sum over all clusters. Next, let $S_{\min}$ be the sum of the m smallest distances among all pairwise distances in the dataset, and let $S_{\max}$ be the sum of the m largest distances. The C-index is then defined [5] as

$$C = \frac{S - S_{\min}}{S_{\max} - S_{\min}}.$$

By construction, the C-index takes values between 0 and 1. Smaller values indicate better clustering quality, since they imply that the within cluster distances are closer to the smallest distances that could theoretically be achieved. Larger values indicate weaker cluster compactness.

The C-index is included because it provides a relative assessment of within-cluster cohesion based on the empirical distance distribution. Unlike the Dunn index, it does not rely only on extreme minimum or maximum distances, but instead evaluates the overall quality of the within-cluster distances in relation to the full set of pairwise dissimilarities. This makes it a useful additional diagnostic when comparing candidate values of K.

The C-index is not, however, entirely free of small-sample effects. It depends on the number of within-cluster pairs m, which becomes small when clusters are small or when a partition contains many tiny clusters. For small m, the reference sums $S_{\min}$ and $S_{\max}$ are each formed from only a few of the smallest and largest distances, so the index is estimated from limited information and can become unstable, and it is undefined in the degenerate case where every cluster contains a single skill ($m = 0$). For this reason the C-index is used alongside the Silhouette and Dunn indices rather than on its own, so that the three criteria together compensate for one another's weaknesses.

4.5 Stability Analysis

Because clustering solutions may be sensitive to small perturbations in the underlying distance matrix, a stability analysis is performed to assess the robustness of the candidate skill groupings. The goal is to determine whether a clustering method yields essentially the same partition when the pairwise distances are subject to minor random variation. A stable clustering solution is

desirable in this context, since operational skill groupings should not change substantially due to small fluctuations in the estimated similarity structure.

Let $D = (d_{ij})_{i,j=1}^{n}$ denote the symmetric distance matrix for the L skills under consideration. For a given clustering method and its tuning parameter (for example, the number of clusters K for k-medoids and spectral clustering, or a distance threshold for hierarchical clustering), let $\mathcal{C}(D)$ denote the partition obtained from the original distance matrix. To evaluate stability, the matrix D is repeatedly perturbed by adding small random noise.

More precisely, for each repetition $r = 1, \ldots, R$, a random symmetric noise matrix

$$E^{(r)} = (e_{ij}^{(r)})_{i,j=1}^{n}$$

is generated, where the entries are sampled from a Gaussian distribution with mean zero and standard deviation σ_ε,

$$e_{ij}^{(r)} \sim \mathcal{N}(0, \sigma_\varepsilon^2).$$

In the implementation used in this thesis, $\sigma_\varepsilon = 0.01$. Symmetry is then enforced by replacing the noise matrix with the average of itself and its transpose,

$$E^{(r)} \leftarrow \tfrac{1}{2}\left(E^{(r)} + (E^{(r)})^\top\right), \qquad e_{ii}^{(r)} = 0,$$

where $\leftarrow$ denotes assignment (the matrix on the right replaces $E^{(r)}$) and $(E^{(r)})^\top$ is the transpose of $E^{(r)}$, obtained by swapping its rows and columns. Averaging a matrix with its transpose makes it symmetric, so that the perturbation applied to the distance between skills i and j is identical to that applied between j and i, as required for a valid distance matrix. The diagonal entries are set to zero, $e_{ii}^{(r)} = 0$, so that the perturbation does not introduce any nonzero self-distance.

The perturbed distance matrix is defined as

$$D^{(r)} = D + E^{(r)}.$$

Since distances must remain non negative, any negative entries produced by the perturbation are truncated to zero. For each perturbed matrix $D^{(r)}$, the clustering algorithm is applied again, yielding a new partition $\mathcal{C}(D^{(r)})$.

The similarity between the original clustering and the clustering obtained from the perturbed matrix is quantified using the Adjusted Rand Index (ARI). Let $z = (z_1, \ldots, z_n)$ denote the cluster labels of the original partition and let $z^{(r)} = (z_1^{(r)}, \ldots, z_n^{(r)})$ denote the labels obtained from the rth perturbed matrix. The Adjusted Rand Index [25], $\mathrm{ARI}(z, z^{(r)})$, measures how well the two partitions agree, while correcting for chance, that is, for the pairs of skills that any two partitions would group together by luck. Its values lie in $[-1, 1]$: a value of 1 means the two clusterings are identical, a value near 0 means the perturbed clustering agrees with the original no more than a random grouping would, and negative values mean the agreement is worse than random.

After R perturbation runs, the stability of a clustering solution is summarized by the sample

mean and sample standard deviation of the ARI values,

$$\overline{\mathrm{ARI}} = \frac{1}{R}\sum_{r=1}^{R} \mathrm{ARI}(z, z^{(r)}),$$

and

$$s_{\mathrm{ARI}} = \sqrt{\frac{1}{R-1}\sum_{r=1}^{R}\left(\mathrm{ARI}(z, z^{(r)}) - \overline{\mathrm{ARI}}\right)^2}.$$

Thus, the implementation computes the average ARI across all perturbation runs together with the sample standard deviation of those ARI values. A clustering solution is considered more stable when it has a high average ARI and a low ARI standard deviation. In this thesis, the procedure is repeated $R = 30$ times for each candidate solution. Following a practical decision rule, solutions with

$$\overline{\mathrm{ARI}} \geq 0.85 \qquad \text{and} \qquad s_{\mathrm{ARI}} \leq 0.15$$

are classified as highly stable. The first condition ensures that the clustering remains almost unchanged under small perturbations, while the second ensures that this robustness is consistent across repetitions.

This perturbation based stability analysis provides an additional diagnostic beyond internal validity measures such as the Silhouette, Dunn, and C-index. Whereas those criteria evaluate the geometric quality of a partition for a fixed distance matrix, the present approach evaluates the sensitivity of the partition to uncertainty in the distance structure itself. As a result, it helps identify clustering solutions that are not only internally well separated, but also robust enough to be meaningful for downstream forecasting and operational decision-making.

4.6 Reconciliation Methods

In a hierarchical forecasting setting, forecasts produced independently at different levels of aggregation are generally not coherent, meaning that skill-level forecasts do not necessarily sum to the corresponding cluster-level forecasts [10]. Reconciliation methods address this by adjusting forecasts across levels so that they satisfy the aggregation constraints implied by the hierarchy. In this thesis, two reconciliation approaches are considered: top-down disaggregation and MinT reconciliation.

4.6.1 Top-Down Disaggregation

The top-down approach generates forecasts at the aggregate level and distributes them to lower-level series using a set of disaggregation proportions [10, 6]. In this thesis, the aggregate level corresponds to data-driven skill clusters, and the lower level corresponds to individual skills within each cluster.

For each cluster c, the individual skill series are aggregated by summing their daily call volumes,

$$Y_t^{(c)} = \sum_{i \in \mathcal{C}_c} x_{t,i}, \tag{4.39}$$

where $\mathcal{C}_c$ is the set of skills in cluster c and $x_{t,i}$ is the call volume for skill i on day t. As at the skill level, every cluster series is treated as multiplicative: the series is log-transformed before fitting an ETS(A,A,A) model and the forecast is back-transformed afterwards, following the $\log(1+x)$ procedure of Section 4.1. This gives an h-step-ahead forecast $\hat{Y}^{(c)}_{T+h}$.

To distribute the cluster-level forecast to individual skills, disaggregation proportions are constructed from historical training data. Following the average historical proportions approach [10, 1], a separate proportion is estimated for each day of the week $d \in \{0, 1, \ldots, 6\}$. Each skill therefore has seven proportions, one per weekday, where the proportion for day d is the skill's share of its cluster's total volume on that weekday (for example, a skill might make up 10% of its cluster's Monday volume but only 2% of its Saturday volume). When a cluster forecast is disaggregated, the skill receives the proportion matching the forecast date's weekday. This captures the strong intra-week seasonality in contact center demand [12], which a single global proportion would ignore.

Volume-adaptive estimation window. The day-of-week proportions are estimated from a recent window of each skill's training data using a trimmed mean, which discards a fraction of the most extreme days before averaging so that spikes do not distort the estimated shares. The window length (how many recent days are used) and the trimming fraction (how aggressively extreme days are removed) are adapted to each skill's demand level, since low- and high-volume queues differ in how noisy and stable their day-of-week behaviour is. Each skill is assigned to a volume category based on the tertiles of the cross-skill distribution of mean daily training volume: let $v_i = \frac{1}{|\mathcal{T}_{\text{train}}|} \sum_{t \in \mathcal{T}_{\text{train}}} x_{t,i}$ be the mean daily training volume of skill i, and $q_{1/3}$, $q_{2/3}$ the lower and upper tertiles of $\{v_i\}$. Skill i is then assigned a window length W_i and a trimming fraction τ_i according to Table 4.1.

Table 4.1: Volume-adaptive window length and trimming fraction used to estimate day-of-week disaggregation profiles. Categories are defined by the tertiles of the cross-skill distribution of mean daily training volume, computed entirely from the training period.

Category	**Volume range**	**Window W_i (days)**	**Trim τ_i**
Low	$v_i < q_{1/3}$	180	0.15
Mid	$q_{1/3} \le v_i < q_{2/3}$	120	0.10
High	$v_i \ge q_{2/3}$	90	0.05

Low-volume skills, which are noisier and contain fewer observations per day-of-week bucket, are assigned the widest window and the heaviest trimming, yielding a stable profile that is robust to sporadic spikes. High-volume skills, which have ample and more regular observations, use a narrower, more responsive window with lighter trimming.

Trimmed-mean day-of-week profile. For each skill i and day-of-week d, let $\mathcal{T}_{i,d}$ be the set of training days with day-of-week d that fall within the most recent W_i days of skill i's training window. Let $n = |\mathcal{T}_{i,d}|$, let $x_{(1)} \le \cdots \le x_{(n)}$ be the corresponding ordered call volumes, and let $g = \lfloor \tau_i n \rfloor$. The historical volume $\mu_{i,d}$ is estimated as the τ_i-trimmed mean of these volumes,

which discards the g smallest and g largest values and averages the remainder,

$$\mu_{i,d} = \frac{1}{n-2g} \sum_{j=g+1}^{n-g} x_{(j)}. \tag{4.40}$$

When fewer than three observations are available in $\mathcal{T}_{i,d}$, no trimming is applied and the plain mean is used. Using a trimmed rather than a simple mean prevents occasional outlier days, such as promotional or incident-driven spikes, from distorting the estimated day-of-week shares.

The disaggregation proportion for skill i on a forecast day with day-of-week d is then

$$p_{i,d} = \frac{\mu_{i,d}}{\sum_{j \in \mathcal{C}_c} \mu_{j,d}}, \tag{4.41}$$

so that $\sum_{i \in \mathcal{C}_c} p_{i,d} = 1$ for every d. When the total cluster volume for a given day-of-week is zero, a uniform fallback $p_{i,d} = 1/|\mathcal{C}_c|$ is applied. The disaggregated forecast for skill i at horizon h is

$$\hat{x}_{T+h,i} = p_{i,\, d(T+h)} \cdot \hat{Y}^{(c)}_{T+h}, \tag{4.42}$$

where $d(T+h)$ denotes the day-of-week of forecast date $T+h$.

The disaggregation scheme described above is a static-proportion top-down method: the proportions are estimated once from the training data and then held fixed across the whole forecast horizon [6]. It remains static even though each skill has seven proportions (one per day of the week) estimated over a volume-adaptive window, because these choices only affect how the proportions are estimated, not whether they change during forecasting; once estimated, the weekday proportions are frozen and simply looked up by the forecast date's day of the week. Static proportions are computationally efficient and perform competitively when lower-level shares are stable over time [10], a reasonable assumption for established contact center queues. The volume-adaptive window and trimmed estimator retain this efficiency while making the shares more robust for the noisier low-volume skills that dominate the portfolio.

4.6.2 Minimum Trace Reconciliation (MinT)

While top-down disaggregation is simple and computationally efficient, it does not make use of the forecast error structure across all levels of the hierarchy. A structural difference is worth noting at the outset. Top-down disaggregation forecasts only the cluster level and obtains skill-level forecasts by splitting these down, so the individual skills are never forecast directly. MinT, by contrast, requires base forecasts at every level of the hierarchy: each cluster and each individual skill is forecast independently, and these forecasts are then reconciled so that they are coherent across levels. Minimum Trace (MinT) reconciliation [30] finds the reconciled forecasts that are unbiased and have minimum variance, exploiting the full covariance structure of the base forecast errors, which is only available because forecasts exist at both levels.

The hierarchical structure of the forecasting problem is expressed through a summing matrix $S \in \mathbb{R}^{n_{\text{all}} \times n_{\text{bottom}}}$, which encodes how the bottom-level series aggregate to upper levels. In this thesis a two-level hierarchy is used, in which the data-driven skill clusters form the upper level

and the individual skills form the bottom level, so that $n_{\text{all}} = n_{\text{clusters}} + n_{\text{skills}}$ and $n_{\text{bottom}} = n_{\text{skills}}$. Let $\hat{\mathbf{y}}_{T+h} \in \mathbb{R}^{n_{\text{all}}}$ denote the vector of base forecasts stacked across both levels, and let $\tilde{\mathbf{y}}_{T+h} \in \mathbb{R}^{n_{\text{bottom}}}$ denote the reconciled bottom-level forecasts. Any coherent reconciliation can be written as

$$\tilde{\mathbf{y}}_{T+h} = P\,\hat{\mathbf{y}}_{T+h}, \tag{4.43}$$

where $P \in \mathbb{R}^{n_{\text{bottom}} \times n_{\text{all}}}$ is a projection matrix. The full reconciled vector across both levels is then recovered as $S\tilde{\mathbf{y}}_{T+h}$.

MinT selects P to minimize the trace of the covariance matrix of the reconciled forecast errors, subject to the unbiasedness constraint $PS = I$. The optimal solution is [30]

$$P = \left(S^\top W^{-1} S\right)^{-1} S^\top W^{-1}, \tag{4.44}$$

where $W \in \mathbb{R}^{n_{\text{all}} \times n_{\text{all}}}$ is the covariance matrix of the h-step-ahead base forecast errors. The reconciled bottom-level forecasts are then

$$\tilde{\mathbf{y}}_{T+h} = \left(S^\top W^{-1} S\right)^{-1} S^\top W^{-1}\,\hat{\mathbf{y}}_{T+h}. \tag{4.45}$$

In practice, W is unknown and must be estimated from in-sample residuals. Let $\hat{e}_t \in \mathbb{R}^{n_{\text{all}}}$ denote the vector of in-sample one-step-ahead forecast errors at time t, computed as the difference between the observed values and the fitted values of the ETS model at each level. Two estimators of W are considered in this thesis.

First, the sample covariance estimator uses the empirical covariance of the in-sample residuals,

$$\hat{W}_{\text{sample}} = \frac{1}{|\mathcal{T}_{\text{train}}|} \sum_{t \in \mathcal{T}_{\text{train}}} \hat{e}_t \hat{e}_t^\top, \tag{4.46}$$

where $|\mathcal{T}_{\text{train}}|$ is the number of days in the training period. This estimator captures the full cross-series error correlation structure but can be unstable when the number of series n_{all} is large relative to $|\mathcal{T}_{\text{train}}|$ [30].

Second, the shrinkage estimator applies Ledoit-Wolf regularization [17] to the sample covariance matrix, shrinking it towards a structured diagonal target to improve the condition number and numerical stability. This estimator is particularly appropriate in the present setting, where the number of skills is large and the training period may not be sufficiently long to reliably estimate all pairwise error covariances.

Because the additive aggregation constraints of the hierarchy hold on the original call-volume scale rather than on the log scale, reconciliation is performed on the original scale: the per-level ETS(A,A,A) base forecasts described in Section 4.1 are fitted on the log-transformed series and back-transformed before reconciliation, and the in-sample residuals used to estimate W are likewise computed on the original scale. Since call volumes are non-negative but the unconstrained MinT solution may return negative skill-level forecasts, non-negativity is enforced directly within the reconciliation rather than by post-hoc clipping. The reconciled bottom-level forecasts are obtained as the solution of the non-negative least-squares problem

$$\tilde{\mathbf{y}}_{T+h} = \arg\min_{\mathbf{y} \geq \mathbf{0}} \left\| W^{-1/2} \left(S\mathbf{y} - \hat{\mathbf{y}}_{T+h}\right) \right\|_2^2, \tag{4.47}$$

where $\|\cdot\|_2$ denotes the Euclidean (ℓ_2) norm of a vector, that is, the square root of the sum of its squared entries, and the outer superscript 2 squares this norm, so that the objective is the sum of squared, covariance-weighted differences between the aggregated reconciled forecasts $S\mathbf{y}$ and the base forecasts $\hat{\mathbf{y}}_{T+h}$. This yields forecasts that are simultaneously coherent and non-negative, while reducing to the closed-form MinT solution when the non-negativity constraint is inactive. The two W estimators are evaluated separately, yielding two MinT variants that are compared against the top-down result.

Chapter 5

Results

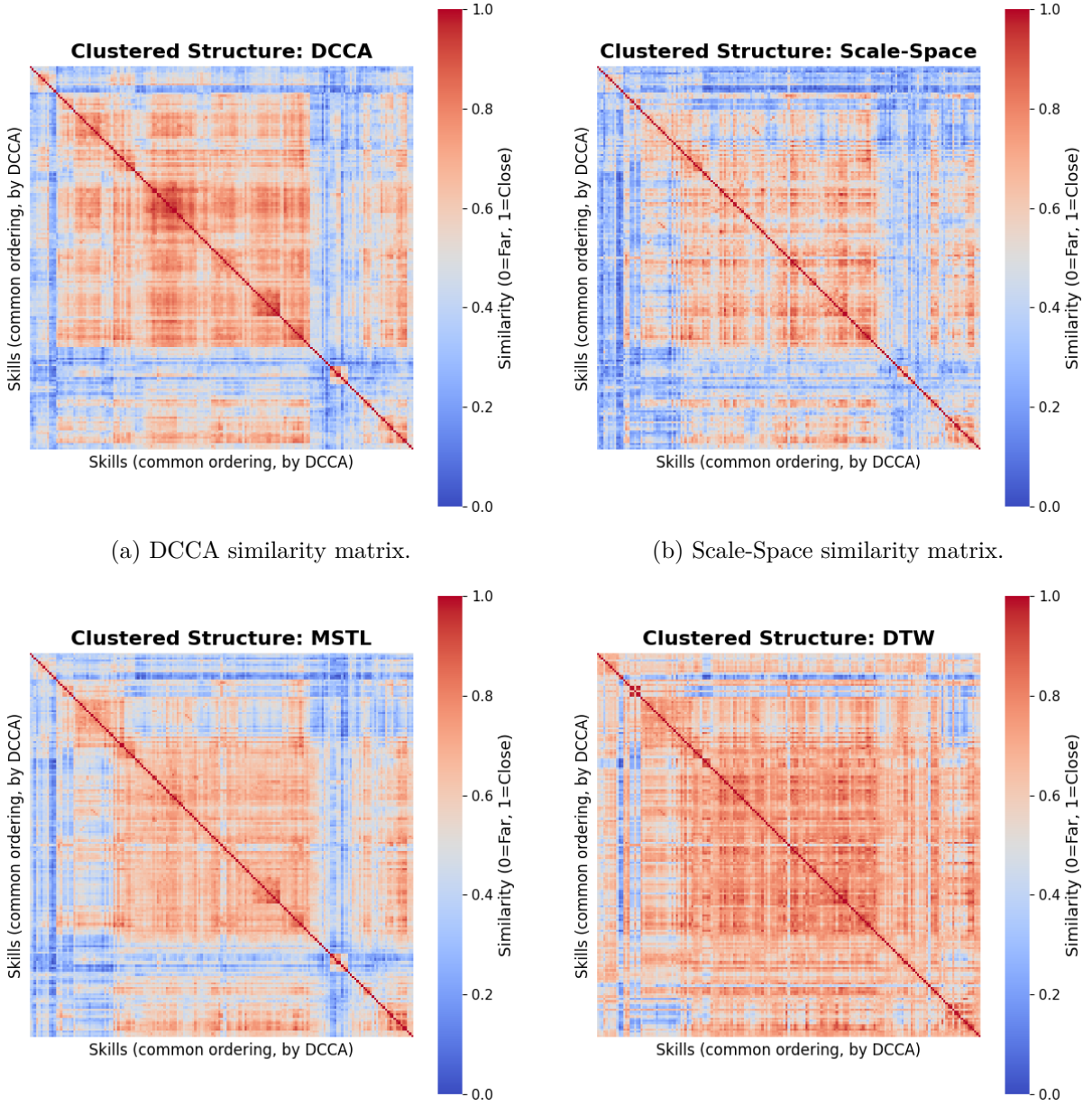
This chapter presents the results of the pipeline introduced in Chapter 4. Section 5.1 reports the four similarity and distance measures and the feature-based distance matrix. Section 5.2 presents the clustering results and identifies the stable skill groupings, and Section 5.2.3 examines the structure of these stable solutions. Section 5.3 evaluates the top-down disaggregation forecasts against the individual-skill baseline. Section 5.4 does the same for MinT reconciliation and compares the two approaches directly. Finally, Section 5.5 explores a per-skill hybrid of the two.

5.1 Results of Similarity and Distance Measures

This section presents the results of the four pairwise similarity and distance measures computed across the 175 skills: DCCA, Scale-Space, MSTL, and DTW. For each measure, the full 175×175 pairwise similarity matrix was computed and visualised as a clustered heatmap with an accompanying dendrogram. The feature-based distance matrix is kept separate and used as an independent clustering track.

5.1.1 Individual similarity matrices

Figure 5.1 presents the similarity heatmaps for DCCA, Scale-Space, MSTL, and DTW. The skill ordering is the same in all four panels, so the same position corresponds to the same pair of skills, allowing the measures to be compared directly. Warm colours indicate high similarity and cool colours low similarity.



(a) DCCA similarity matrix.

(b) Scale-Space similarity matrix.

(c) MSTL similarity matrix.

(d) DTW similarity matrix.

Figure 5.1: Clustered similarity heatmaps for all four individual measures. Skills follow the same ordering across different similarity measures. Red indicates high similarity, close to 1, and blue indicates low similarity, close to 0.

The DCCA heatmap in Figure 5.1a shows a broad block structure: a large group of similar skills and a smaller group that is more weakly connected to the rest. This fits the purpose of DCCA, which captures long-range co-movement after removing local fluctuations. In a call centre, many skills are driven by the same organisation-wide factors, such as weekly cycles and long-term trends, so DCCA groups together the skills that move together over time and separates those with a more distinct trajectory. Because the shared ordering is derived from DCCA, this structure appears most cleanly on its diagonal and serves as the reference layout for the other three panels.

The Scale-Space heatmap in Figure 5.1b largely preserves the same core and margins under

the shared ordering, but with a finer, more textured pattern within the warm core. This is because Scale-Space compares skills across multiple time scales: two skills can be similar in their long-run trend but different in their short-term movements, or the reverse, so it sees additional fine-grained structure that long-run co-movement alone does not separate.

The MSTL heatmap in Figure 5.1c closely resembles the DCCA and Scale-Space panels under the shared ordering, which is consistent with their high rank agreement. The main visible difference is that the cool margins are lighter and less extensive: fewer skill pairs fall into the strongly dissimilar (blue) range, giving a slightly warmer overall appearance. This reflects the strong weekly seasonality found in the exploratory analysis, since the shared day-of-week pattern raises the baseline similarity of many skill pairs once the series are decomposed. The panel still shows clear subgroups and cooler regions, so differences in trend and residual behaviour remain informative even when the weekly pattern is common. Among the four time-series measures, MSTL retains the most discriminating signal, because decomposing the series lets the trend and residual components distinguish skills even when the weekly seasonal component is shared. This is confirmed by the clustering results in Section 5.2, which also show that the feature-based representation, not any time-series measure, produces the best-separated and most stable groupings.

The DTW heatmap in Figure 5.1d stands out from the other three: it is uniformly warm and shows little block structure under the shared ordering. This is because DTW measures similarity differently, comparing series after flexible time alignment so that it is sensitive to similar demand shapes even when peaks occur a little earlier or later. Its similarities therefore do not line up with the co-movement-based DCCA ordering, which is the visual counterpart of its low rank agreement with the other measures. This makes DTW complementary rather than redundant, as it can group skills with similar shapes that do not coincide exactly in calendar time, which is why it is included in the consensus matrix after normalisation so that its shape-based signal contributes alongside the others.

Figure 5.2 shows the distribution of off-diagonal pairwise similarity scores for DCCA, Scale-Space, MSTL, and DTW. DCCA, Scale-Space, and MSTL all produce broadly concentrated distributions centred around moderate similarity values. MSTL is the most concentrated, with a sharper peak around the middle of the similarity range, reflecting the common seasonal structure shared by many skills. DCCA and Scale-Space are slightly wider than MSTL, indicating more variation in long-run and multi-scale similarity across skills.

DTW has the widest and most distinct distribution: its scores spread across nearly the whole range, with many skill pairs scored as very similar and many as very dissimilar, rather than clustering around a moderate value like the other three measures. This confirms that DTW judges similarity differently, by demand shape under flexible time alignment rather than by co-movement. Because of this, DTW is kept in the consensus but on an equal footing with the others: the distances are normalised before averaging, so its broad spread does not let it dominate the combined matrix. DTW therefore adds a complementary, shape-based view of similarity rather than overriding the co-movement-based measures.

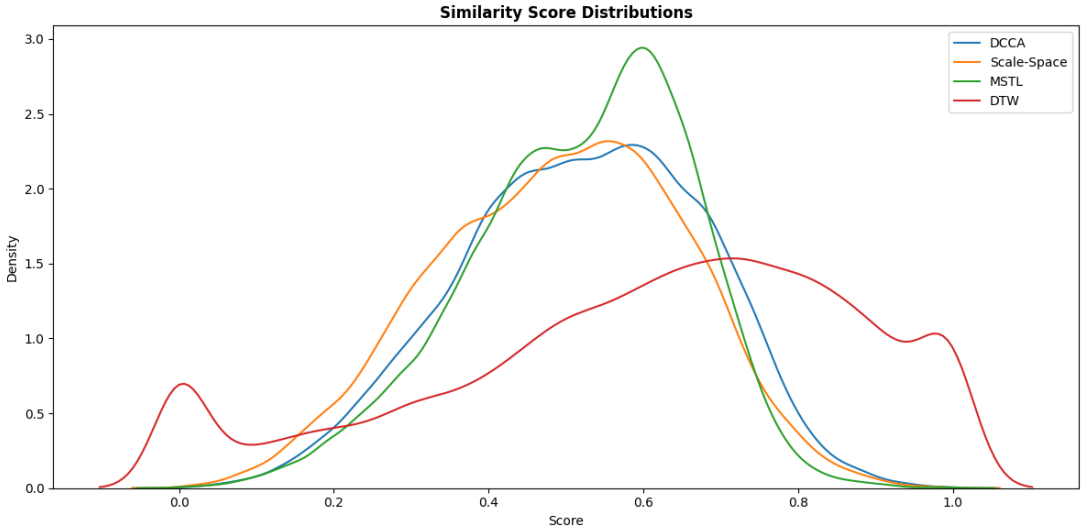


Figure 5.2: Kernel density estimates of the off-diagonal pairwise similarity scores for DCCA, Scale-Space, MSTL, and DTW. DCCA, Scale-Space, and MSTL are concentrated around moderate similarity values, while DTW has a wider distribution, reflecting its different sensitivity to time-aligned demand-shape similarity.

5.1.2 Agreement Between Similarity Measures

The four measures each capture a different notion of similarity, so it is useful to check whether they still agree on which skill pairs are similar. This is hard to see from the heatmaps, so it is measured directly. For each pair of measures, the off-diagonal entries of the two similarity matrices were correlated across all $\binom{175}{2} = 15{,}225$ skill pairs using Spearman's rank correlation ρ_s [4], which compares how the pairs are ranked from least to most similar without assuming the four scales are directly comparable. The results are shown in Figure 5.3.

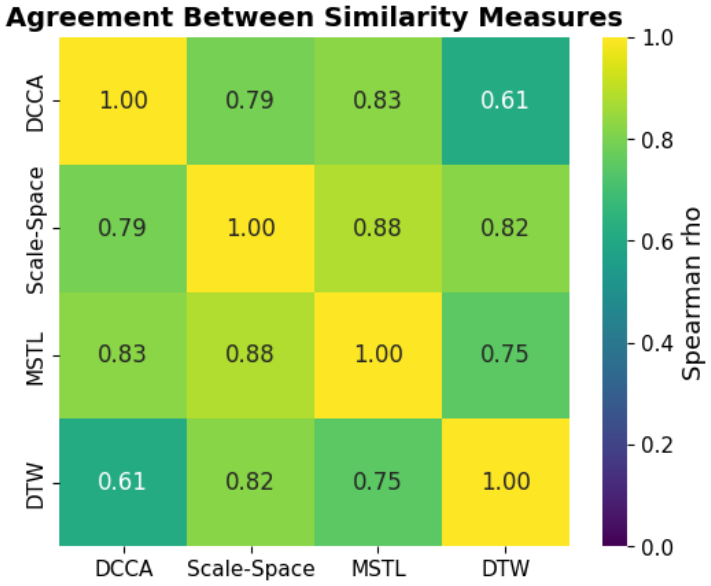


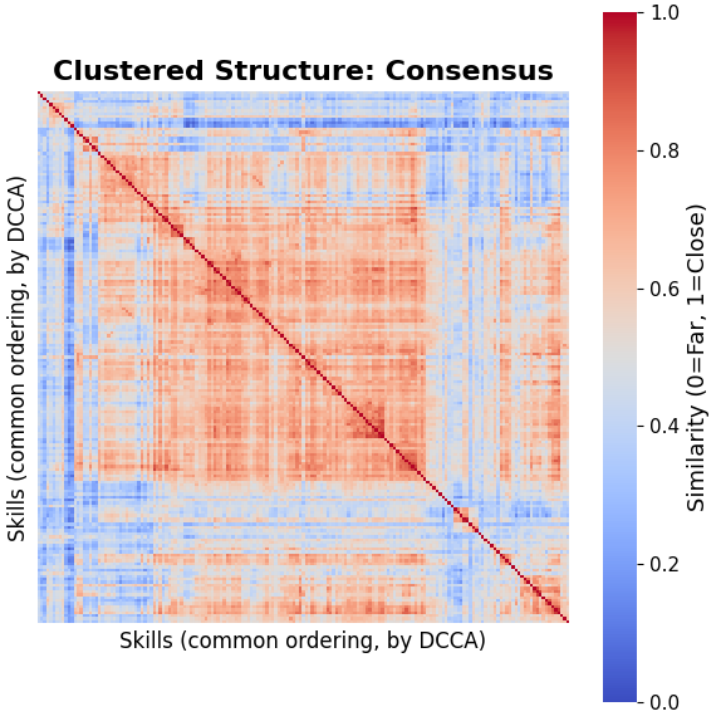
Figure 5.3: Spearman rank correlation between the off-diagonal entries of the four similarity matrices, over all $\binom{175}{2}$ skill pairs. Higher values mean stronger agreement on which pairs are ranked as similar.

All four measures agree strongly, with every correlation at least 0.61. The three correlation- and decomposition-based measures are the most alike, with DCCA, Scale-Space, and MSTL agreeing at ρ_s between 0.79 and 0.88. DTW agrees least, and its lowest value is with DCCA ($\rho_s = 0.61$): these are the two most opposite measures, since DCCA looks at long-range co-movement and DTW at time-aligned demand shape. DTW still agrees more with Scale-Space ($\rho_s = 0.82$), which sits between the two.

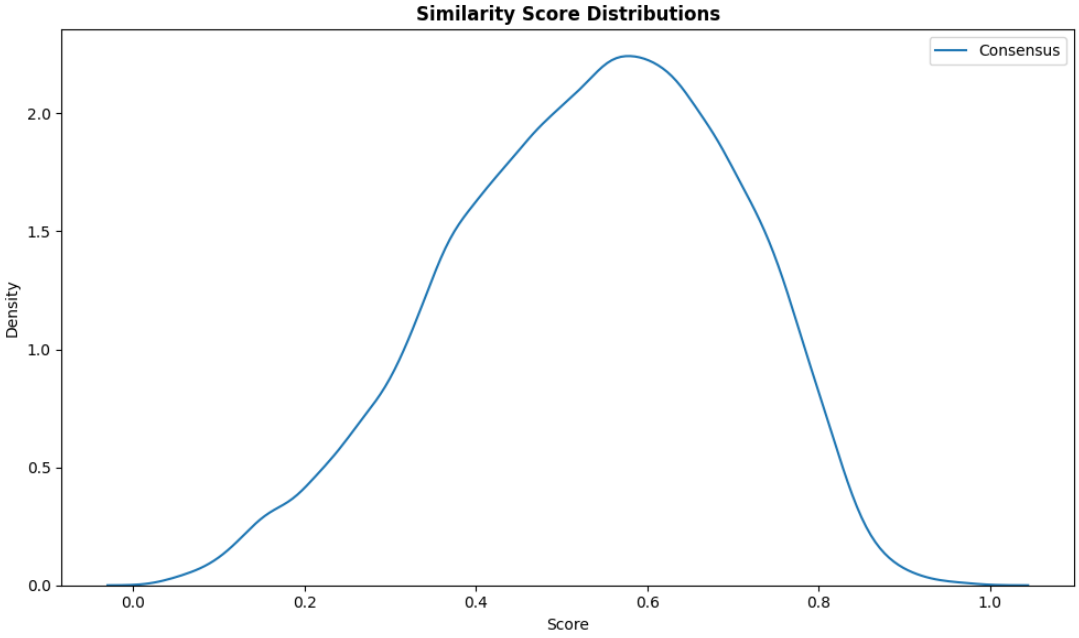
This level of agreement is what makes combining the measures useful: the correlations are high enough that averaging them gives a coherent consensus, but low enough that each measure still adds something the others miss. The raw scores also show that DTW gives higher similarity values than the other three, which is why its heatmap (Figure 5.1d) looks warmer overall.

5.1.3 Consensus similarity matrix

The consensus matrix is constructed by averaging the normalised distance matrices of DCCA, Scale-Space, MSTL, and DTW. Including all four measures allows the consensus to combine complementary views of similarity. DCCA captures long-range co-movement, Scale-Space captures correlation across multiple temporal resolutions, MSTL captures similarity in decomposed trend and seasonal components, and DTW captures similarity in demand shape under flexible time alignment. Since each matrix is normalised before averaging, no single measure dominates solely because of its scale or score distribution.



(a) Clustered consensus similarity heatmap.



(b) Score distribution of the consensus matrix.

Figure 5.4: Consensus similarity matrix constructed by averaging the normalised distance matrices of DCCA, Scale-Space, MSTL, and DTW. The heatmap shows the combined clustered structure across all four similarity perspectives, while the density plot shows the distribution of consensus similarity scores.

Figures 5.4a and 5.4b show the consensus similarity matrix and its score distribution. Under the shared ordering, the heatmap has a clear clustered structure: a large warm block of mutually similar skills occupies the centre of the matrix along the diagonal, indicating that many skills share common demand dynamics, while cooler blue bands along the margins mark a smaller

group of skills that are only weakly similar to the rest and stand out as separate. Between these, some skills have intermediate similarity to the main block.

Compared with the individual heatmaps, the consensus matrix gives a balanced view of similarity. It keeps the broad co-moving structure from DCCA and MSTL, the more detailed subgroups from Scale-Space, and the sharper shape-based separations from DTW, so it is more robust than any single measure alone. The score distribution is unimodal, with a peak around 0.6 and a gradual tail toward lower values, showing that the averaging produces a stable, moderately high similarity signal rather than a fragmented or dominated one.

The consensus matrix is therefore used as the main similarity input for clustering. It combines several complementary notions of similarity while reducing the dependence on any single assumption, which matters here because skills differ not only in volume but also in trend, seasonal strength, and the timing of demand changes.

5.1.4 Feature-based distance matrix

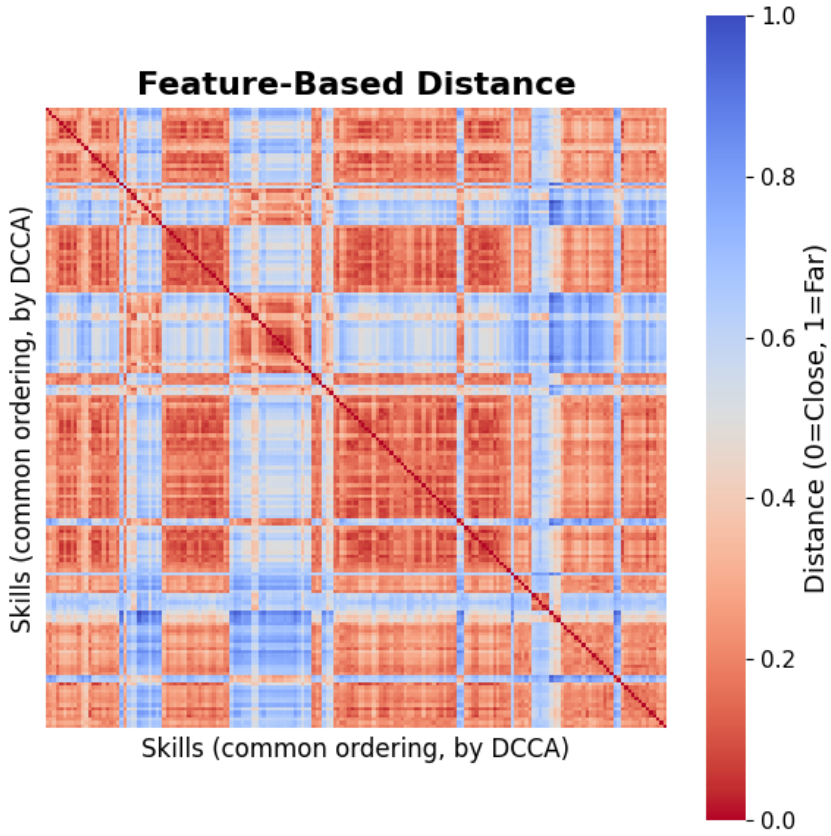
Figures 5.5a and 5.5b show the feature-based distance matrix and its distribution. Unlike the time-series matrices, this one uses a distance scale: white cells are skills that are nearly identical in their features, and dark blue cells are skills that are very different. The matrix is shown in the same ordering as the previous heatmaps, so it can be compared with them directly.

The clearest feature of the heatmap is the crosshatch of blue vertical and horizontal bands. These are skills that are far from almost all other skills on every feature at once. They are not just a few isolated points but a fairly large group of skills that behave differently from the rest of the portfolio. One possible reason, which we did not check against the individual skills, is that these are unusual queues, for example with very high or low weekend activity, very irregular demand, or day-of-week patterns unlike the others; confirming this would mean looking at the skills one by one. Away from these bands, most cells are red or pale, which means that the remaining skill pairs are either close or only moderately different.

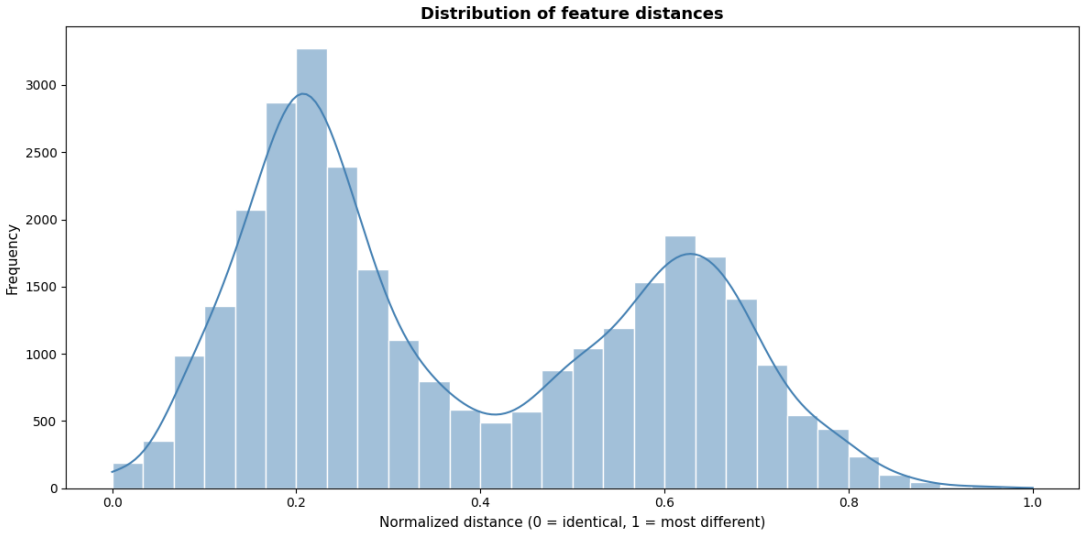
The distance distribution in Figure 5.5b shows the same structure more clearly. It has two peaks: a main one around 0.2 and a second one around 0.6, with a dip near 0.4 between them and a long tail towards 0.9. The first peak is the large group of similar skill pairs, the second is a sizeable group of clearly different pairs, and the tail comes from the unusual skills seen as bands in the heatmap. The portfolio is therefore not one uniform group, but two broad groups of skill pairs plus a separate set of unusual skills.

The feature-based matrix gives a genuinely different view from the time-series measures. Instead of asking whether two skills move together over time, it asks whether they have a similar operational profile: relative volatility, trend, weekend activity, additive or multiplicative character, and day-of-week shape. Because the features are standardised and absolute volume is left out, two skills with very different sizes but a similar shape are treated as close, while two skills that move together can still be far apart if their variability or weekly profile differ. This bimodal, clearly separated structure shows that the feature space contains real structure: the skills do not form one uniform cloud but fall into groups that differ in their operational behaviour. This is easier to interpret than the time-series measures, because each feature has a direct meaning. For example, two skills are close in feature space when they have a similar share

of weekend demand, a similar level of volatility, and a similar day-of-week shape, regardless of how large they are or whether they move together over time. A pair of skills can therefore be close here even if a correlation measure treats them as different, and vice versa. Whether this clearer structure also leads to better clusters and forecasts is examined in the clustering and forecasting results; there, the feature-based representation is kept as a separate clustering track alongside the consensus matrix.



(a) Feature-based distance matrix in original skill ordering.



(b) Distribution of normalised feature distances.

Figure 5.5: Feature-based distance matrix (left) and its score distribution (right). Note that this matrix uses a distance scale, where white indicates identical skills (distance = 0) and dark blue indicates maximum dissimilarity (distance = 1). The crosshatch pattern of dark lines reveals a small number of structurally atypical skills that are highly dissimilar from almost all others. The distance distribution is bimodal, with a primary mode around 0.2 and a pronounced secondary mode around 0.6, separated by a trough near 0.4, and a tail extending toward 0.9.

5.1.5 Why the Measures Differ in Separating Skills

The clustering and stability results presented later in Sections 5.2 show a clear split between the representations: only the feature-based and MSTL measures produced clusters that were both well separated and stable, while DCCA, Scale-Space, and DTW did not. This subsection explains why, drawing the results together with the seasonality analysis in Section 3.3.3. The starting point is that almost all skills (99.4%) share the same strong day-of-week cycle. When similarity is measured directly on the raw series, this shared weekly pattern makes most pairs of skills look alike, leaving little signal to tell them apart.

The two correlation-based measures and the shape-based one are each dominated by this shared cycle, though for slightly different reasons. DCCA removes local trends before measuring co-movement, but it does not remove the weekly cycle; once the trend is gone, the main remaining signal is the weekly pattern that almost all skills share, so their detrended fluctuations move together and the similarities are uniformly high. Scale-Space measures correlation at several time scales, but because the weekly cycle is strong and common to nearly every skill, the correlation is high at the relevant scales for almost all pairs, so the different scales do not separate the skills here. DTW compares series after flexible time alignment, applied with a seven-day warping band; this lets two skills be aligned across a full week, so their weekly peaks are matched regardless of which weekday they fall on. This removes much of the day-of-week information that actually distinguishes skills. As a result, none of these three measures produced stable, well-separated clusters.

MSTL performs better because it first splits each series into trend, seasonal, and remainder components and compares them separately, giving the shared seasonal component only one third of the weight. Even though the seasonal part is common to most skills, the trend and remainder parts still differ, so MSTL keeps enough signal to separate skills, although only at a coarse level.

The feature-based representation separates skills most clearly because it works in a different way: instead of asking whether two skills move together over time, it describes each skill by its operational profile, in particular the shape of its day-of-week demand. This distinction matters, because peaking on the same day is not the same as having the same weekly shape. Two skills that both dip at weekends look almost identical to a correlation measure, since they move in the same direction at the same time, yet their profiles can be very different: one might handle 20% of its volume at weekends and the other none at all. The feature-based measure captures exactly this difference. Because it describes how demand is spread across the days of the week, and leaves out the absolute volume level, it separates skills that the correlation measures treat as similar, which is why it produces the clearest and most stable groupings.

5.2 Clustering Results and Model Selection

This section presents the results of the clustering stage, in which hierarchical clustering, k-medoids, and spectral clustering were applied to each of the five distance matrices: DCCA, Scale-Space (SS), MSTL, Consensus, and Features, for values of k ranging from 2 to 25. The upper bound of $k = 25$ was chosen to ensure that clusters remain statistically meaningful for 175 skills, preventing configurations that are too granular to support reliable forecasting. For each

combination of distance matrix, clustering method, and value of k, three internal validity indices were computed: the Silhouette index (higher is better), the Dunn index (higher is better), and the C-index (lower is better), as defined in Section 4.4.

5.2.1 Validation procedure

A per-method composite-rank procedure was used to identify robust candidate skill groupings. For each clustering method, all configurations were pooled across the five input representations (DCCA, Scale-Space, MSTL, DTW, and the feature-based distance matrix) where a configuration is defined by the combination of input matrix, requested number of clusters, and resulting (actual) number of clusters. Configurations with missing validity scores were excluded.

Within each method, the three validity metrics were combined into a single composite rank. Each configuration was ranked separately on the three metrics, in descending order for Silhouette and Dunn, which are maximised, and in ascending order for the C-index, which is minimised, so that rank one always denotes the best value on a given metric, and the three ranks were averaged to obtain the composite rank. A low composite rank therefore indicates a configuration that performs consistently well across all three metrics simultaneously, rather than excelling on one metric while performing poorly on the others. The ten configurations with the lowest (best) composite rank were retained for each clustering method. The per-method shortlists were then combined to form the overall shortlist, which contains the ten best-ranked configurations from each clustering method and is presented in Table 5.1.

Table 5.1 contains only feature-based and MSTL configurations; no DCCA, Scale-Space, DTW, or consensus configuration ranked in the top ten for any method. As explained above, this is because the strong weekly cycle shared by almost all skills makes the correlation- and shape-based measures treat most pairs as similar, leaving too little signal to form well-separated clusters. The feature-based measure (which compares day-of-week shape) and MSTL (which gives the shared seasonal component only one third of the weight) are the only measures that separate skills clearly, which is why they dominate the shortlist.

The shortlist also indicates what range of cluster counts is appropriate for this data. Across all three methods, the best composite ranks are found at small k: hierarchical clustering favours k between 2 and 7 (composite ranks 2.0–5.0), while k-medoids and spectral clustering rank $k = 2$ and $k = 3$ highest. The validity indices then deteriorate steadily as k increases, and no configuration with k above about 7 reaches the top of the shortlist. This suggests that the data support only a small number of well-defined groups, and that larger values of k tend to fragment the portfolio rather than reveal genuine additional structure, a point examined further in Section 5.2.3.

The absolute level of the scores tells a similar story. Only the low-k feature-based and (for k-medoids and spectral) the very top configurations reach clearly good values, with Silhouette scores around 0.5–0.6, high Dunn values, and near-zero C-index. Beyond these, the scores drop quickly: most configurations in the table have a Silhouette below 0.2, which indicates weak separation, with many skills lying close to the boundary between clusters. In other words, only a handful of configurations, concentrated at small k and almost all feature-based, are genuinely well separated; the rest appear in the shortlist only because they are the best of a weak field

within their method, not because they form strong clusters in absolute terms. This reinforces the choice to carry only the strongest candidates forward and to confirm their quality with the stability analysis in Section 5.2.2.

Matrix	k	Silhouette ↑	Dunn ↑	C-index ↓	Composite rank ↓
Hierarchical					
Features	5	0.5388	0.5674	0.0034	**2.00**
Features	4	0.5986	0.4956	0.0094	2.33
Features	3	0.5978	0.4277	0.0125	3.67
Features	7	0.4965	0.4517	0.0042	3.67
Features	6	0.4988	0.4053	0.0048	4.67
Features	2	0.5816	0.4145	0.0481	5.00
Features	21	0.2196	0.1988	0.0476	11.33
Features	18	0.2106	0.1854	0.0530	13.67
Features	11	0.2298	0.1515	0.0603	15.67
Features	10	0.2334	0.1440	0.0606	15.67
K-Medoids					
Features	2	0.5812	0.3413	0.0532	**1.33**
Features	4	0.5288	0.0536	0.0254	21.00
Features	21	0.1515	0.0619	0.0705	22.67
Features	3	0.2435	0.0772	0.1518	22.83
Features	12	0.1455	0.0630	0.0950	23.00
MSTL	19	0.0947	0.1337	0.1572	26.00
Features	5	0.1541	0.0772	0.1609	29.17
Features	24	0.1173	0.0590	0.1039	32.00
Features	13	0.1029	0.0605	0.1120	32.33
MSTL	13	0.0858	0.1650	0.1705	32.33
Spectral					
Features	2	0.5879	0.4039	0.0254	**1.33**
Features	3	0.5539	0.2405	0.0116	1.67
Features	6	0.2179	0.0775	0.1177	12.00
MSTL	4	0.1727	0.1084	0.1632	16.67
MSTL	3	0.1750	0.0938	0.1962	21.00
Features	7	0.2005	0.0522	0.0885	24.33
Features	9	0.1378	0.0555	0.1257	26.17
Features	5	0.2319	0.0466	0.1051	27.00
Features	12	0.0858	0.0677	0.1159	27.00
Features	8	0.1436	0.0527	0.1222	27.67

Table 5.1: Shortlisted clustering configurations: the ten configurations with the lowest composite validity rank within each clustering method. For the Silhouette and Dunn indices higher values are better (↑) and for the C-index lower values are better (↓); the composite rank averages the three per-metric ranks, so a lower value is better. The requested and actual numbers of clusters coincided for all shortlisted configurations. The best configuration per method is shown in bold.

5.2.2 Stability analysis

Validity metrics measure how well-defined a clustering solution is on the original data, but they do not indicate whether the same grouping would emerge if the data were slightly perturbed. A clustering solution that changes substantially under small changes to the distance matrix is not reliable enough to use as the basis for further analysis. To assess this, a stability test was applied to the shortlisted configurations of Table 5.1 using the Adjusted Rand Index (ARI).

For each configuration, the clustering algorithm was first run on the original distance matrix to obtain a reference set of cluster assignments. Small random noise was then added to the distance matrix, drawn from a normal distribution with mean zero and standard deviation 0.01. The perturbed matrix was made symmetric and clipped to ensure that all distances remained non-negative, after which the clustering algorithm was run again. The ARI between the original and perturbed assignments was used to measure stability. As defined in Section 4.5, an ARI of 1.0 means the perturbed grouping is identical to the original, while a value near 0 means it agrees with the original no more than a random grouping would. This procedure was repeated 30 times per configuration. A configuration was considered stable if it achieved a mean ARI of at least 0.85 and a standard deviation of at most 0.15.

Table 5.2: Final stable clustering configurations retained after the perturbation analysis, with the mean and standard deviation of the Adjusted Rand Index (ARI) over 30 perturbations. All retained configurations satisfy the stability criteria of mean ARI ≥ 0.85 and standard deviation ≤ 0.15.

Matrix	**Method**	k	**Mean ARI**	**Std ARI**
Features	Hierarchical	2	1.000	0.000
Features	Hierarchical	3	1.000	0.000
Features	Hierarchical	4	1.000	0.000
Features	Hierarchical	5	1.000	0.000
Features	Hierarchical	6	0.990	0.009
Features	Hierarchical	7	0.988	0.007
Features	Spectral	2	1.000	0.000
Features	Spectral	3	0.997	0.003
Features	Spectral	5	0.905	0.104
MSTL	Spectral	3	0.991	0.032
MSTL	Spectral	4	0.975	0.026

Eleven configurations passed the stability thresholds, all from the feature-based or MSTL representation and using either hierarchical or spectral clustering. The feature representation dominates, with nine of the eleven (six hierarchical, three spectral); the other two are MSTL spectral solutions. Across the retained configurations the mean ARI ranged from 0.905 to 1.000, all comfortably above the 0.85 threshold.

No K-medoids configuration passed the stability test, so the table contains only hierarchical and spectral solutions. K-medoids is fragile for skills that lie between clusters. A boundary skill is one that is almost equally close to its own medoid and to a neighbouring cluster's medoid, so

a small amount of added noise can flip it to the other cluster. When it flips, it can change that cluster's medoid, which in turn makes further skills flip, so the whole partition ends up different. Stability also degrades at high k, because forcing the algorithm to produce more groups than naturally exist in the data makes the extra clusters unstable.

A small number of clusters is appropriate here rather than a limitation. The validity indices already pointed to few well-defined groups, and the feature space consists of one large core of skills that behave alike, plus a few atypical skills, so there are simply not many distinct groups to find. Grouping many skills together is also the intended effect: the aim is to pool skills that behave alike so that the combined series is smoother and easier to forecast than the individual skills. The risk of using too few clusters, that genuinely different skills are merged, is limited in this dataset because the clusters remain homogeneous even at low k, and the forecasting results in Section 5.3 confirm that the low-k feature-based solutions are also the most accurate. For a different contact centre with more heterogeneous demand, more clusters might be needed, so the appropriate number should be re-assessed per dataset rather than fixed in advance.

The feature-based configurations are the most robust. Features Hierarchical passed at every shortlisted granularity, $k \in \{2, 3, 4, 5, 6, 7\}$, and Features Spectral for $k \in \{2, 3, 5\}$. The hierarchical solutions for $k \in \{2, 3, 4, 5\}$ achieved a mean ARI of exactly 1.0 (unchanged across all 30 perturbations), with $k = 6$ and $k = 7$ almost perfectly stable (≈ 0.99). Features Spectral was similarly robust for $k = 2$ (1.0) and $k = 3$ (0.997), while $k = 5$ was the least stable retained configuration (mean ARI 0.905, std 0.104), still within the criteria but noticeably less stable. This confirms that the feature representation produces robust cluster assignments across a range of cluster counts.

Among the time-series representations, only MSTL produced stable partitions (MSTL Spectral for $k \in \{3, 4\}$, mean ARI 0.991 and 0.975), with slightly higher standard deviations (around 0.03) than the strongest feature-based solutions. DCCA, Scale-Space, DTW, and the consensus matrix contributed no stable configuration; none even reached the validity shortlist, so they were not carried into the perturbation analysis.

Overall, the stability analysis narrows the shortlist to a smaller set of robust candidates, most credibly the feature-based hierarchical and spectral configurations. These form the final candidate set carried forward into the forecasting evaluation.

5.2.3 Cluster Structure of the Final Stable Solutions

Figures 5.6 to 5.16 present the cluster separation diagnostics for all eleven stable configurations, one figure per configuration. Each shows a Silhouette plot (left), a within- versus between-cluster distance distribution (centre), and a t-SNE projection coloured by cluster assignment (right), used here purely for visualisation [24].

Feature-based configurations: The nine feature-based configurations give the strongest separation across all three diagnostics, with mean Silhouette scores from 0.232 (Features Spectral $k = 5$) to 0.599 (Features Hierarchical $k = 4$), within-cluster distances clearly below between-cluster distances, and barely overlapping t-SNE groups. They all share the same uneven cluster sizes: one large core of about 125 skills, a second group of about 42–47, and one or more very

small groups. This is because the feature space contains real structure, a dense core plus a few scattered outliers (the crosshatch skills with extreme volatility, unusual weekend activity, or atypical shapes). As k increases, the hierarchical solutions simply peel these outliers off into small clusters while the core stays the same, which is why the diagnostics barely change for $k \in \{2, \ldots, 7\}$. Separation is best at low granularity (Features Hierarchical $k = 4$ and $k = 3$, around 0.60), while Features Spectral $k = 5$ is weakest (0.232), as it cuts the core into medium-sized groups rather than peeling off outliers.

MSTL Spectral $k = 3$ and $k = 4$: The two MSTL configurations separate much less well (mean Silhouettes 0.175 and 0.173), with within- and between-cluster distances overlapping heavily and interleaved t-SNE groups. This is because the MSTL distances contain no clear structure: with 99.4% of skills sharing the weekly cycle, almost all pairs are moderately similar and there are no outliers to peel off. MSTL therefore just slices this even spread into balanced groups (sizes 100/46/29 at $k = 3$; 91/46/28/10 at $k = 4$), rather than separating a core from outliers. The result is less sharply separated but more even in size, which is operationally useful when balanced, forecastable groups are preferred over maximally distinct ones.

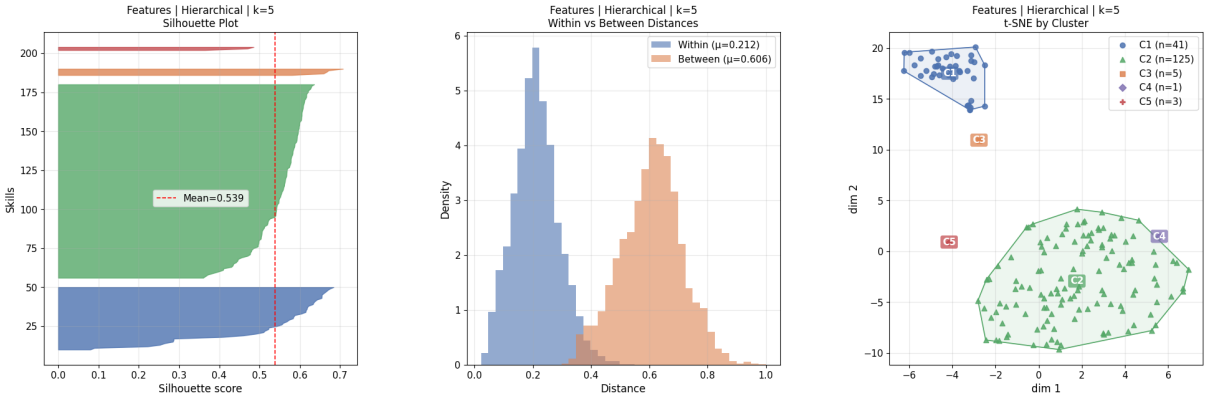


Figure 5.6: Cluster separation quality diagnostics for Features Hierarchical $k = 5$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

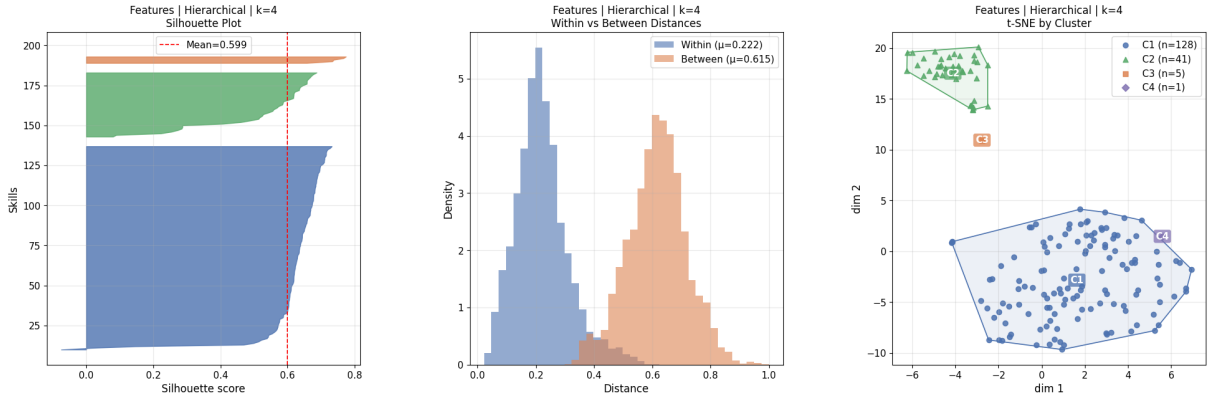


Figure 5.7: Cluster separation quality diagnostics for Features Hierarchical $k = 4$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

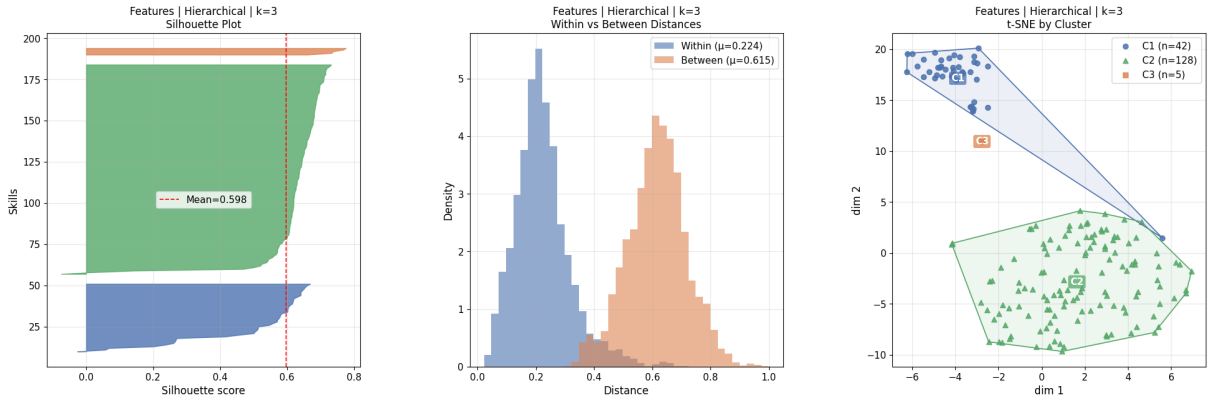


Figure 5.8: Cluster separation quality diagnostics for Features Hierarchical $k = 3$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

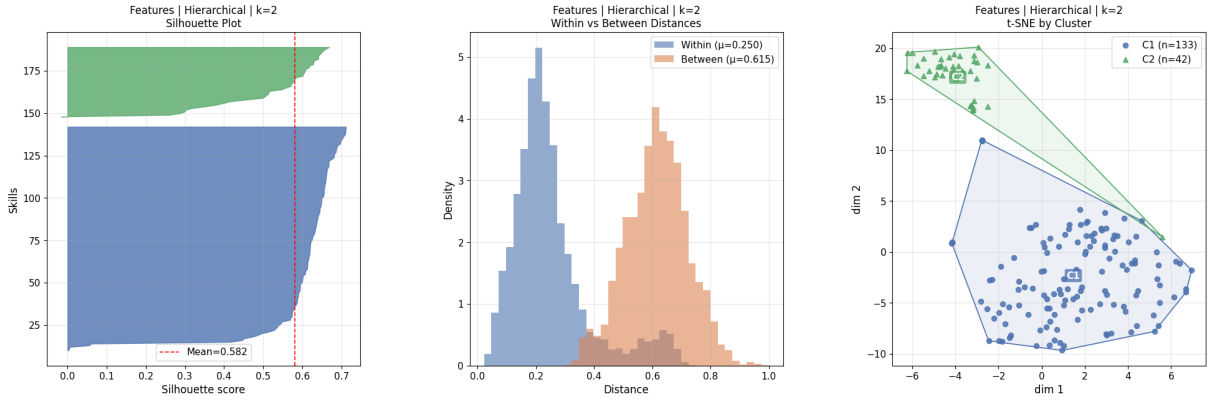


Figure 5.9: Cluster separation quality diagnostics for Features Hierarchical $k = 2$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

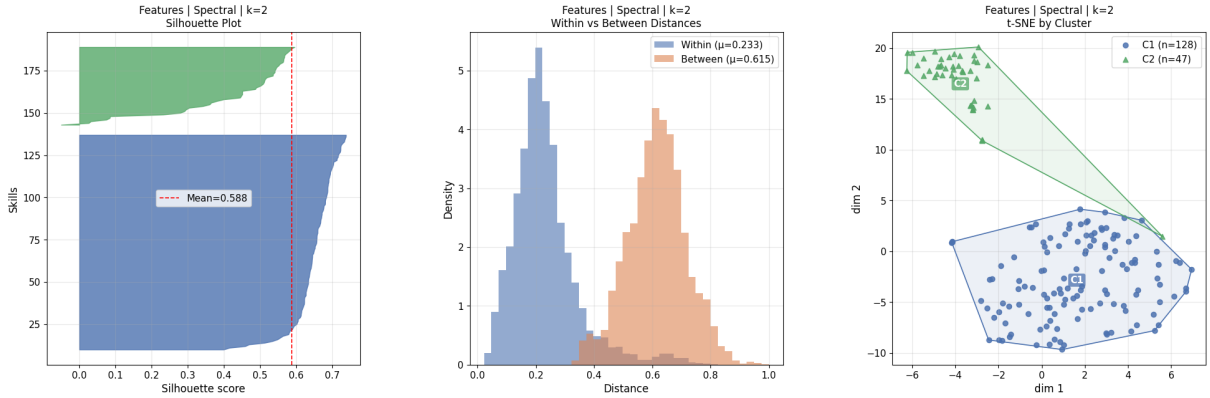


Figure 5.10: Cluster separation quality diagnostics for Features Spectral $k = 2$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

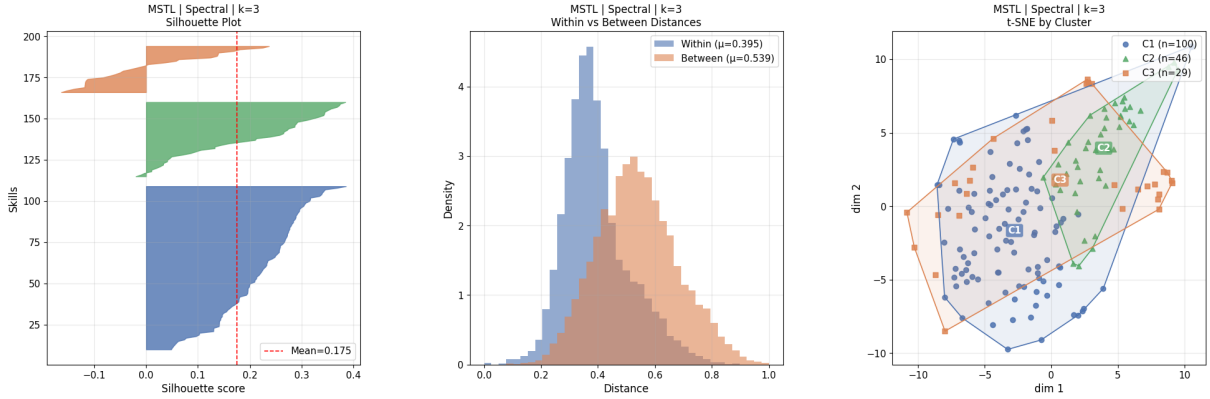


Figure 5.11: Cluster separation quality diagnostics for Features Spectral $k = 3$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

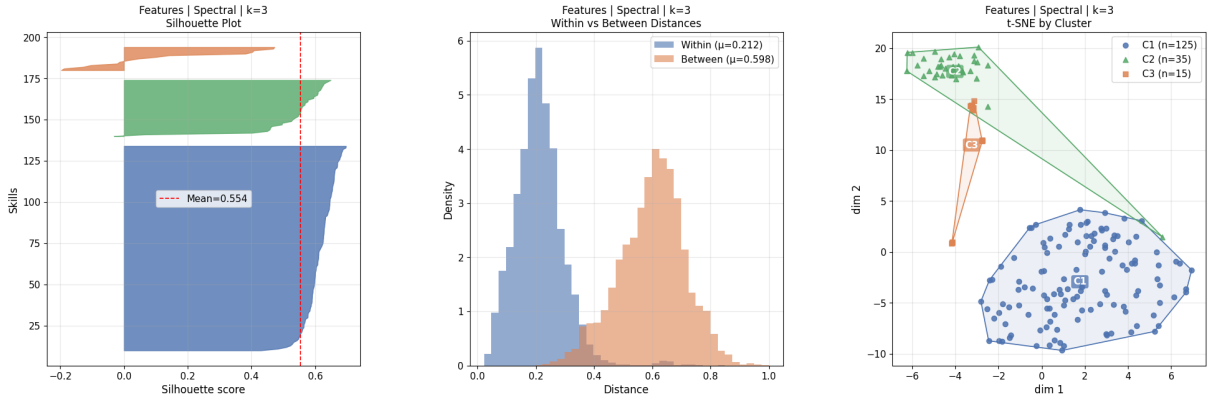


Figure 5.12: Cluster separation quality diagnostics for Features Hierarchical $k = 6$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

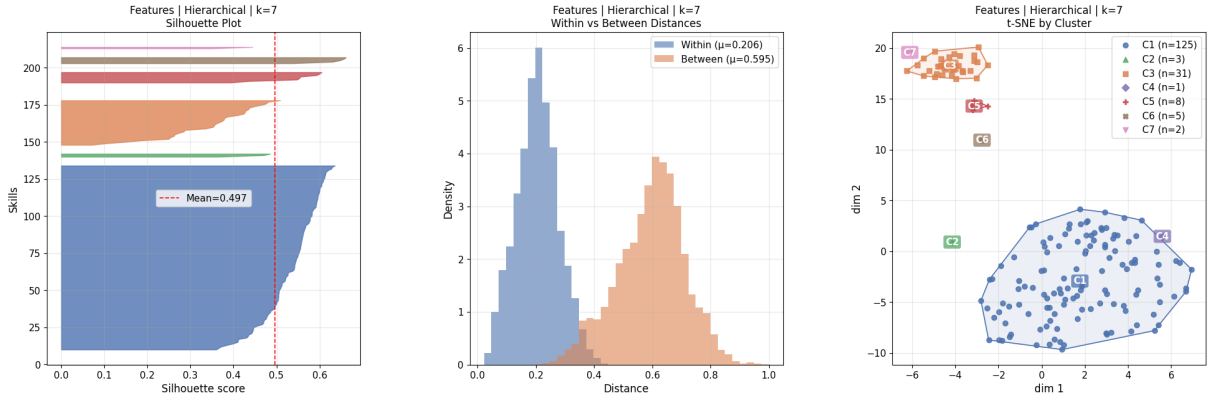


Figure 5.13: Cluster separation quality diagnostics for Features Hierarchical $k = 7$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

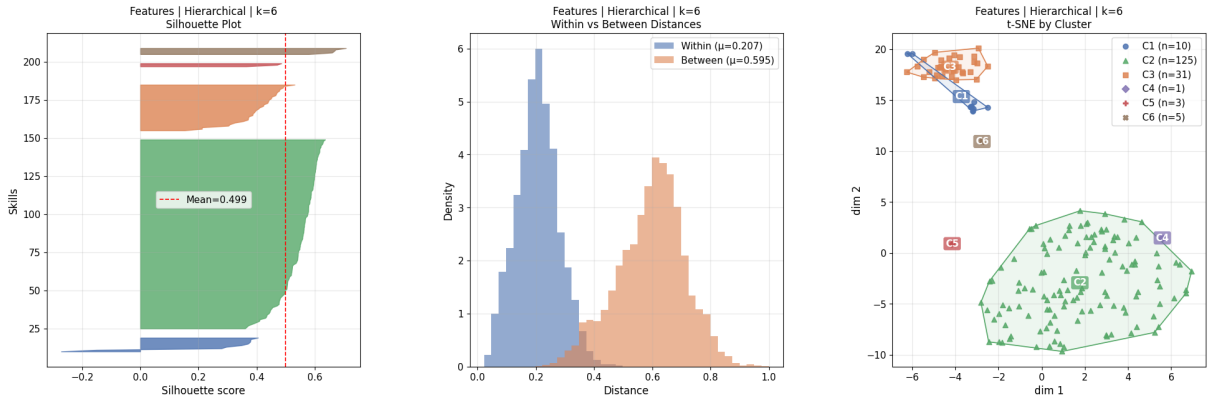


Figure 5.14: Cluster separation quality diagnostics for Features Spectral $k = 5$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

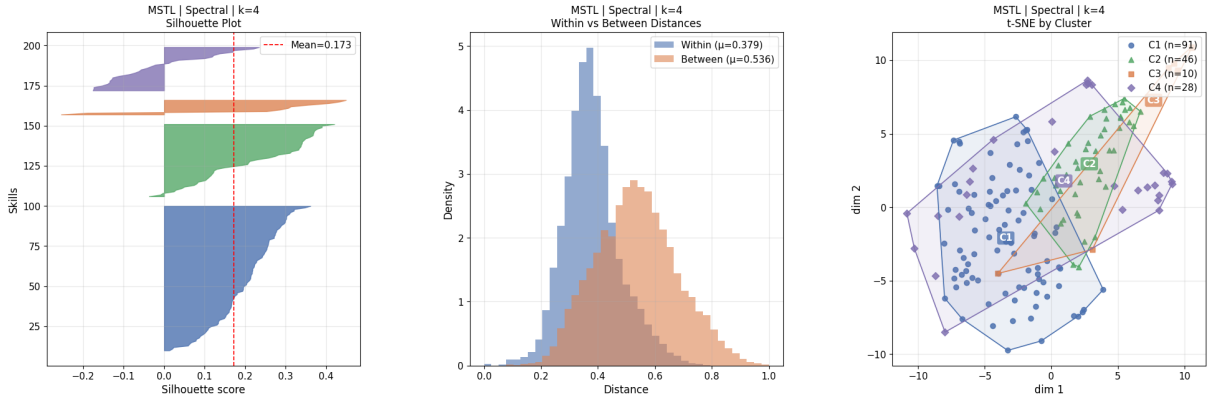


Figure 5.15: Cluster separation quality diagnostics for MSTL Spectral $k = 3$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

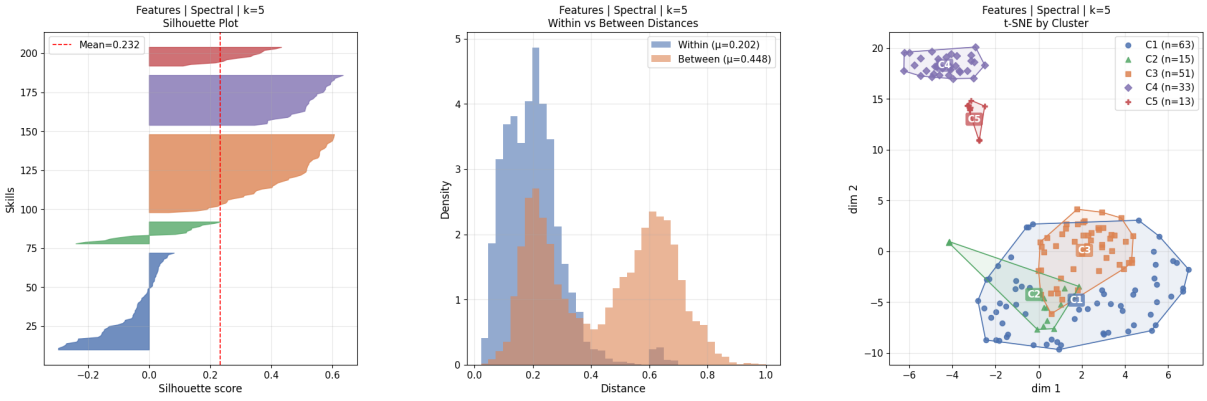


Figure 5.16: Cluster separation quality diagnostics for MSTL Spectral $k = 4$. Left: Silhouette plot with the mean score (red dashed line). Centre: within-cluster (blue) and between-cluster (orange) pairwise distances, with means (μ) in the legend. Right: t-SNE projection of the 175 skills coloured by cluster.

5.3 Top-Down Forecasting Results

This section presents the results of the top-down disaggregation forecasting approach applied to the eleven stable clustering configurations. For each configuration, a cluster-level ETS model was fitted on the aggregated call volume series and forecasts were disaggregated to individual skill level using the volume-adaptive, trimmed day-of-week proportions described in Section 4.6.1. Forecasts were evaluated over the test period spanning 19 August 2025 to 16 June 2026, corresponding to the final 30% of the observation window, while all models and proportions were estimated solely on the first 70% of the data. The resulting skill-level forecasts were evaluated using the system Sum of Absolute Errors (SAE), aggregated across all 175 skills, and compared against an individual skill baseline in which a separate ETS model was fitted for each skill independently. Throughout this section, improvement is reported as the percentage reduction in system SAE relative to this baseline; because the total actual volume is identical for the baseline and the disaggregated forecasts, this relative reduction is equivalent to the corresponding reduction in a volume-weighted error such as WAPE.

5.3.1 Baseline

The individual skill baseline achieves a system SAE of 781,936. This is the reference point against which all cluster-based top-down forecasting results are assessed, and it corresponds to the partner company's current operational practice, in which each skill is forecast individually and no skills are merged; outperforming it therefore directly addresses whether data-driven grouping improves on current practice. To give a sense of scale, this corresponds to a mean per-skill WAPE of roughly 35%, meaning the average skill is mis-forecast by about 35% of the volume it handles over the test period.

This error level is expected rather than a shortcoming, because the forecasting model was deliberately kept simple: the focus of this thesis is the clustering and reconciliation, not the forecasting model. A Holt-Winters ETS was chosen for its interpretability and robustness,

using only the call-volume history and a weekly seasonal cycle, with a per-skill additive or multiplicative-log choice. It does not use the business knowledge that operational forecasts normally rely on, such as public holidays, closed days, campaigns, or outages, which would lower the absolute error but are outside the scope of this study. Since the same model is used for both the baseline and the cluster-level forecasts, any improvement reported below comes from the clustering and disaggregation rather than the forecasting model.

5.3.2 Overall Forecasting Performance

All eleven configurations beat the individual baseline (Table 5.3), so cluster-level top-down forecasting consistently improves system-level accuracy. The improvement ranges from 13.63% (MSTL Spectral $k = 3$) to 22.94% (Features Spectral $k = 2$, the lowest SAE at 602,580). The three best configurations, Features Spectral $k = 2$, Features Spectral $k = 3$, and Features Hierarchical $k = 2$, are within 0.15 percentage points and are effectively tied. By construction, the disaggregated skill forecasts sum exactly to their cluster forecast, since the day-of-week proportions sum to one, so the approach is coherent across both levels without further adjustment.

There is a clear gap between feature-based and MSTL configurations: the nine feature-based ones reduce the error by 18.14%–22.94%, while the two MSTL ones reduce it by only 14.54% and 13.63%. . This is driven by cluster quality. Feature-based clustering groups skills by operational behaviour (volatility, trend, weekend activity, day-of-week profile), so skills in a cluster behave alike and their combined series is smooth and easy to forecast. MSTL groups skills mainly by their shared weekly cycle, so cluster members are not really more similar to each other than to skills in other clusters; even so, they still give a clear gain of around 14%. The feature-based hierarchical configurations with $k \in \{2, 3, 4, 5\}$ are a good practical choice, since they give almost the same accuracy (22.3–22.8%) with a simpler, more interpretable method. Overall, feature-based grouping is the most effective approach, and cluster quality is the main driver of the improvement.

Table 5.3: Top-down disaggregation forecasting results for all eleven stable configurations, ranked by system SAE. The individual baseline system SAE is 781,936 for all configurations. Improvement is the percentage reduction in system SAE relative to the baseline; the final column reports how many of the 175 skills improved under disaggregation.

Matrix	Method	k	System SAE	Improvement (%)	Skills improved
Features	Spectral	2	602,580	22.94	100
Features	Spectral	3	603,209	22.86	106
Features	Hierarchical	2	603,589	22.81	102
Features	Hierarchical	3	606,757	22.40	100
Features	Hierarchical	4	607,401	22.32	100
Features	Hierarchical	5	607,408	22.32	100
Features	Hierarchical	7	609,115	22.10	100
Features	Hierarchical	6	609,548	22.05	101
Features	Spectral	5	640,061	18.14	100
MSTL	Spectral	4	668,210	14.54	97
MSTL	Spectral	3	675,348	13.63	98

To see how this gain is spread across individual skills, Figure 5.17 shows the per-skill percentage improvement in SAE for each configuration, defined for each skill as $(\mathrm{SAE}_{\mathrm{Individual}} - \mathrm{SAE}_{\mathrm{Disaggregated}})/\mathrm{SAE}_{\mathrm{Individual}} \times 100$, so that positive values indicate that disaggregation outperforms the individual baseline. Expressed per skill, where every skill counts equally regardless of size, the improvement is much smaller than the system-level numbers suggest. For every configuration the median improvement is only a few percent (roughly 3–6%), and the inter-quartile range straddles zero, so a substantial share of skills are forecast slightly worse and a similar share slightly better. The whiskers are long and roughly symmetric (about ±45–55%). For the typical skill, therefore, top-down disaggregation makes little difference, improving or worsening the forecast only marginally.

The gap between the system-level reduction and the per-skill boxplot shows where the aggregate gain comes from. The system SAE reduction of about 22% is volume-weighted and dominated by high-volume skills, whereas the boxplot gives every skill equal influence and shows only a small central improvement. These two views are consistent only if the large, consistent error reductions are concentrated in the high-volume skills, while most low- and mid-volume skills change little. High-volume skills benefit most because pooling them into a cluster smooths their combined series and stabilises the day-of-week structure that drives most of their volume, and because their large absolute errors dominate the system metric. Most of the reduction in system SAE therefore comes from the high-volume skills, even though the median skill barely changes.

The ordering of the boxes broadly follows the system-level ranking, with the feature-based configurations sitting slightly higher than the MSTL ones on the per-skill median. The better-structured clusters therefore help the typical skill a little more, not only the high-volume ones. The effect is small, however, and should not be over-interpreted.

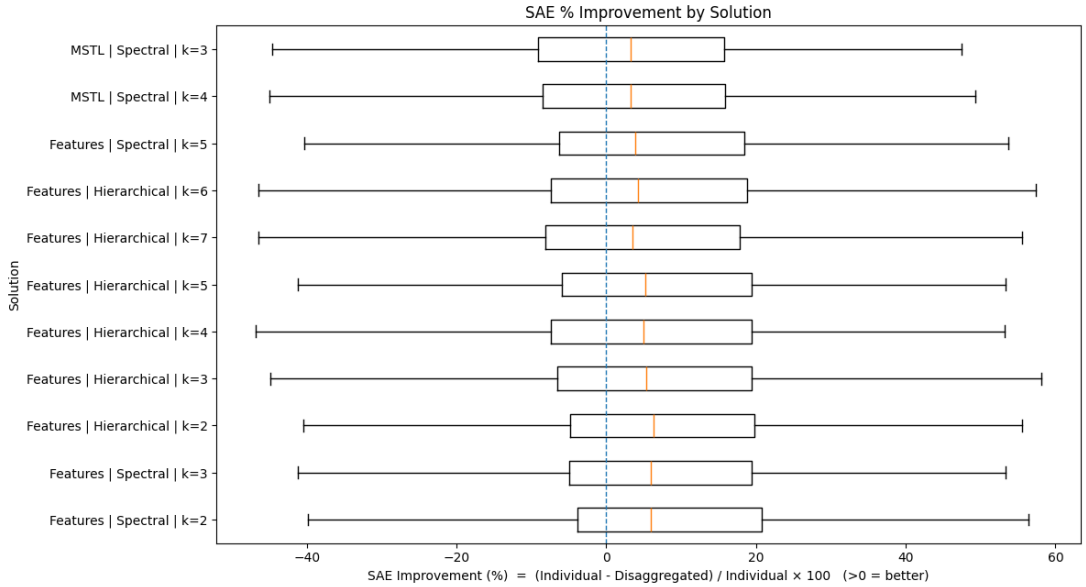


Figure 5.17: Distribution of the per-skill SAE percentage improvement, for each stable configuration; positive values indicate that disaggregation outperforms the individual baseline. Medians lie just above zero (around 3–6%) and the inter-quartile ranges straddle zero, with long, roughly symmetric whiskers. Because each skill is weighted equally, the typical per-skill improvement is small even though the volume-weighted system SAE falls by up to 22.94%, confirming that the aggregate gain is concentrated in the high-volume skills.

5.3.3 Illustrative Example

To make the effect of clustering and disaggregation concrete, Figure 5.18 compares the individual and disaggregated forecasts for Skill_178, a high-volume skill under the best configuration (Features Spectral $k = 2$). Over the test period the skill's demand declines steadily from its training level. The individual ETS model anchors on the higher level seen during training and continues to forecast large weekly peaks of around 350 calls, well above the actual demand, giving a system SAE of 28,518. The disaggregated forecast instead inherits the smoother, cluster-level aggregate, which captures the overall downward drift shared across the cluster, and reapplies the skill's stable day-of-week proportions; it therefore tracks the declining actual much more closely and roughly halves the error to 14,237. The disaggregated forecast still over-forecasts somewhat in the later part of the test window, because the day-of-week proportions are fixed and cannot adapt to how quickly this particular skill declines, but it is far closer than the individual model throughout. This single skill illustrates the mechanism behind the system-level gains: pooling cleans and stabilises the signal at the cluster level, and the day-of-week proportions carry that cleaner signal back down to the skill.

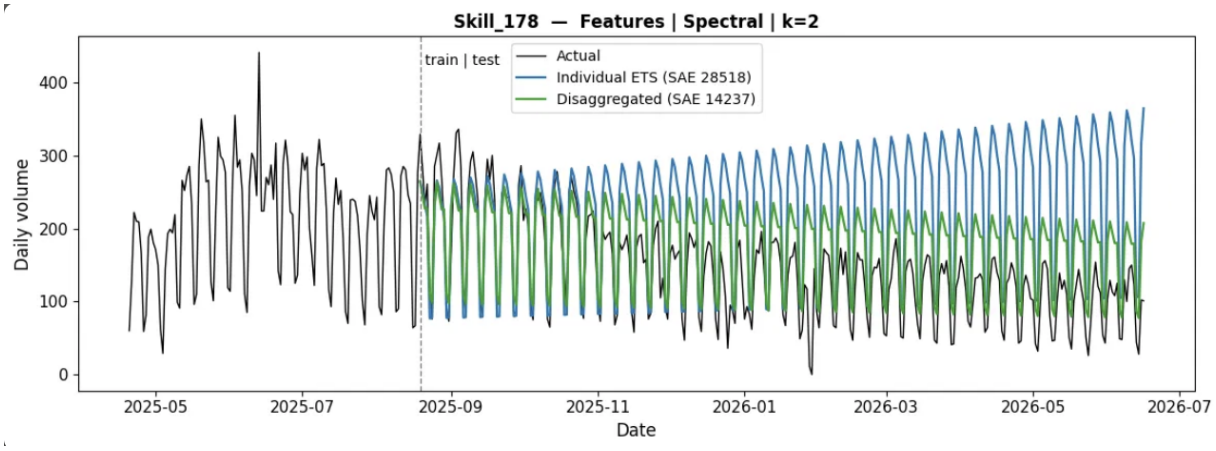


Figure 5.18: Individual versus disaggregated forecast for Skill_178 (a high-volume skill) under Features Spectral $k = 2$. The individual ETS forecast (blue) over-forecasts the declining demand, while the disaggregated forecast (green) inherits the smoother cluster trend and tracks the actual (black) more closely, roughly halving the SAE. The dashed line marks the train/test split.

5.3.4 Performance across volume groups

To understand where the improvement comes from, the per-skill results were split into three equally sized volume groups: low (59 skills), mid (58 skills), and high (58 skills). Table 5.4 reports, for each group, the pooled percentage SAE reduction and the percentage of skills improved.

Table 5.4: Top-down disaggregation by volume group, for each stable configuration. For each group: the pooled percentage SAE reduction (% red.) and the percentage of skills improved (% impr.). Positive reductions indicate improvement over the individual baseline.

			Low		Mid		High	
Matrix	Method	k	% red.	% impr.	% red.	% impr.	% red.	% impr.
Features	Spectral	2	2.3	53	21.5	55	24.1	64
Features	Spectral	3	1.1	59	21.2	59	24.1	64
Features	Hierarchical	2	2.4	54	21.5	57	23.9	64
Features	Hierarchical	3	2.1	54	20.2	53	23.6	64
Features	Hierarchical	4	2.0	54	20.1	53	23.6	64
Features	Hierarchical	5	2.0	54	20.0	53	23.6	64
Features	Hierarchical	7	1.3	58	19.4	52	23.4	62
Features	Hierarchical	6	1.4	58	19.4	53	23.4	62
Features	Spectral	5	−0.3	58	19.9	55	18.6	59
MSTL	Spectral	4	1.0	54	17.8	50	14.6	62
MSTL	Spectral	3	−0.3	51	17.8	52	13.5	66

The results are not what hierarchical pooling is usually expected to deliver. In theory, pooling should help the noisy low-volume skills most, since they are the hardest to forecast alone. In our experiments, the opposite happens: the improvement rises with volume. For

the best configurations the pooled reduction is only about 1–2% for low-volume skills, around 20–21% for mid-volume skills, and roughly 24% for high-volume skills.

The increase in improvement with volume comes from the two steps of top-down disaggregation. Pooling the cluster aggregate helps all skills by smoothing out noise, but the second step, splitting the aggregate back using day-of-week proportions, treats skills very differently. High- and mid-volume skills have many observations per weekday, so their day-of-week proportions are stable, and the smooth, low-noise cluster forecast is preserved when it is split back to the skill level.

Low-volume skills have short, erratic histories, so their proportions are noisy and the split re-adds much of the volatility that pooling had just removed, leaving only about a 2% gain.Interestingly, the fraction of skills that improve is similar across all three groups (around 50–64%), so it is the size of the gain, not the chance of improving, that grows with volume.

A second, separate effect is the metric. SAE is measured in raw calls, so low-volume skills have small errors to begin with and can contribute almost nothing to the system total, regardless of their relative improvement. High-volume skills have large errors, so even a moderate relative gain on them produces a large absolute reduction. In absolute terms, high-volume skills therefore account for roughly 83–87% of the total system improvement in every configuration.

The same feature-versus-MSTL gap seen at the system level reappears within the high-volume group, where the feature-based configurations reduce the error by 23–24% against 13.5–14.6% for the two MSTL configurations. The one exception is Features Spectral $k = 5$, whose high-volume reduction falls to 18.6%, consistent with its weaker cluster separation: at $k = 5$ the spectral split places a few large skills into a cluster that forecasts them worse.

Across all three groups the typical skill is helped slightly rather than harmed, but the largest gains, and hence most of the system-level improvement, are clearly concentrated in the high- and mid-volume skills rather than the low-volume ones.

5.3.5 Performance on sparse skills

The 64 sparse skills removed during preprocessing (Section 3.2) were too incomplete to cluster or model reliably on their own. As an additional test, each skill was assigned to one of the existing clusters and forecasted by disaggregating that cluster's forecast, and the result was compared against forecasting each sparse skill individually. This is a cold-start test of whether data-driven grouping can help skills that cannot be forecast in isolation, so the absolute numbers are not directly comparable to the main results in Table 5.3.

The improvement is very large and consistent across all configurations (Table 5.5): the pooled SAE reduction is around 90% for every configuration (89.8%–92.0%), and a clear majority of sparse skills improve (59%–88%). The pooled reduction barely varies between methods, but the share of skills improved separates them more: the feature-based configurations improve the most skills (76%–88%), with Features Hierarchical $k = 2$ and Features Spectral $k = 2$ the best at 87.5%, while the two MSTL configurations improve the fewest (59%–69%). So the same feature-versus-MSTL ordering seen for the established skills reappears here, although it is now visible in the share of skills improved rather than in the pooled reduction, which is near-saturated for every method.

This very large improvement reflects the weakness of the baseline rather than an unusually good forecast: an individual ETS model fitted to a series with more than half its observations missing produces poor and unstable forecasts, so almost any method that borrows strength from a larger, more complete aggregate beats it by a wide margin. The result still shows an important practical benefit of data-driven grouping: it extends usable forecasts to low-coverage skills that would otherwise be forecast poorly or dropped, which is directly relevant for the partner company, where new and sparse queues appear regularly.

Table 5.5: Application to sparse skills: SAE improvement per stable configuration, pooled across the 64 held-out sparse skills (cold-start rows excluded). Pooled % is the volume-weighted system SAE reduction; % improved is the share of sparse skills forecast better under disaggregation.

Matrix	Method	k	SAE Improvement %	% Skills Improved
Features	Hierarchical	2	90.0	87.5
Features	Spectral	2	89.9	87.5
Features	Spectral	3	89.9	84.4
Features	Hierarchical	3	89.9	82.8
Features	Hierarchical	5	89.9	82.8
Features	Spectral	5	91.0	82.8
Features	Hierarchical	6	89.9	81.3
Features	Hierarchical	4	89.9	78.1
Features	Hierarchical	7	89.8	76.6
MSTL	Spectral	4	91.3	68.8
MSTL	Spectral	3	92.0	59.4

5.3.6 Performance on new skills

The eight new skills identified during preprocessing (Section 3.2) appeared too recently to be included in the clustering or in the main comparison. Six of them have continuous, well-populated coverage over their short lifetime (the fully new skills), while two also have sparse coverage (the new-and-sparse skills). To test whether the data-driven grouping helps recently introduced queues, each new skill was assigned to one of the existing clusters and forecast by disaggregating that cluster's forecast, and the result was compared against forecasting the skill individually from the first date of its own data. So that all eight skills are active over the same period, the comparison is restricted to the window in which their coverage overlaps, 2026-01-30 to 2026-05-21, with the 70/30 train/test split computed inside that window. As with the sparse experiment, this is a cold-start feasibility test rather than a continuation of the main comparison, so the absolute numbers are not directly comparable to the headline results in Table 5.3.

The results are very similar across all eleven stable configurations, so for clarity only one is shown here (MSTL Spectral $k = 3$); the other solutions give the same pattern. In every solution, seven of the eight new skills improve under disaggregation, and the same skill (Skill_273) is the only one that does not. Figure 5.19 shows the per-skill improvement for this configuration, defined for each skill as $(\mathrm{SAE}_{\mathrm{Individual}} - \mathrm{SAE}_{\mathrm{Disaggregated}})/\mathrm{SAE}_{\mathrm{Individual}} \times 100$, so that positive values

indicate that disaggregation outperforms individual forecasting: most skills improve clearly (for example +82%, +55%, and +54% for the three largest gains), while Skill_273 is forecast about 50% worse than its own baseline.

Skill_273 illustrates that placement quality matters. Figure 5.20 shows why it fails: over its short lifetime its demand has a clear upward trend and rises to high peaks, but the cluster it is assigned to does not share this rising level, so the disaggregated forecast (which only rescales the cluster's shape) systematically underforecasts the peaks. Its own ETS model, fitted directly to the rising series, captures the trend better and therefore achieves a lower SAE. Disaggregation is thus beneficial for the large majority of new skills, but not when a skill's own dynamics (here, a strong trend) differ from those of its assigned cluster.

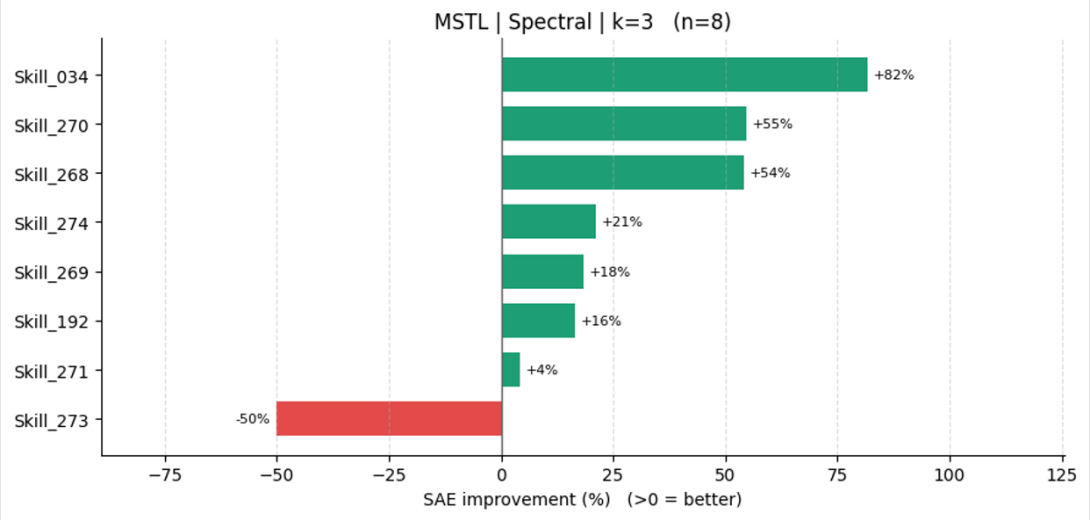


Figure 5.19: Per-skill SAE percentage improvement, for the eight held-out new skills under MSTL Spectral $k = 3$ (representative of all stable configurations). Positive values indicate that disaggregation outperforms individual forecasting. Seven of the eight skills improve; only Skill_273 is forecast worse.

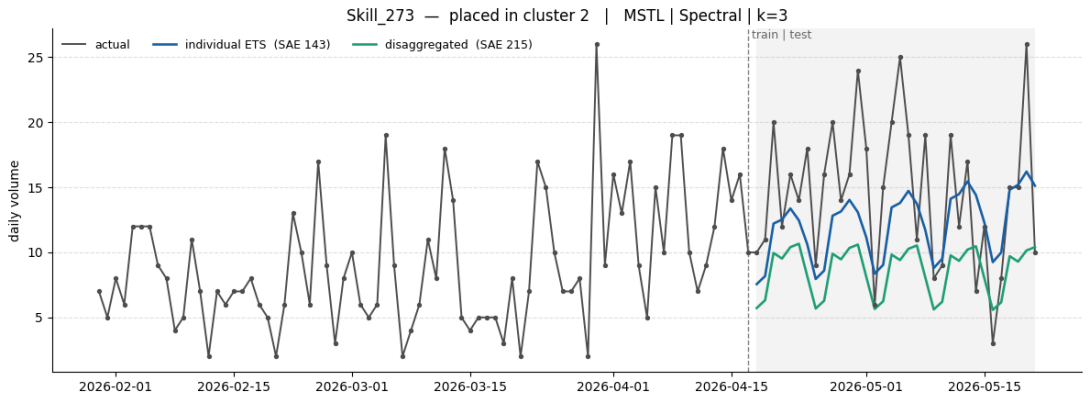


Figure 5.20: Forecast for Skill_273, placed in cluster 2 under MSTL Spectral $k = 3$. The individual ETS forecast (blue, SAE 143) follows the rising level of the series, while the disaggregated forecast (green, SAE 215) inherits the cluster's lower level and underforecasts the peaks. This is the one new skill for which disaggregation does not improve on the individual baseline, illustrating that placement quality matters when a skill's dynamics differ from its cluster's.

5.4 MinT Forecasting Results

This section reports the results of the MinT reconciliation approach applied to the same eleven stable clustering configurations used for top-down disaggregation. For each configuration, base forecasts were generated independently at the cluster and skill levels using the ETS model described in Section 4.1, and reconciled with the two-level summing matrix using the Sample and Shrinkage estimators of the error covariance W. Reconciled forecasts were obtained by non-negative least squares, so that the skill-level forecasts are simultaneously coherent with their cluster and non-negative. Performance is measured by the system Sum of Absolute Errors (SAE) aggregated across all 175 skills and evaluated over the same test period (19 August 2025 to 16 June 2026) and against the same individual-skill baseline (system SAE 781,936) as in Section 5.3, so that the two reconciliation approaches are directly comparable.

Table 5.6 reports the system SAE for all 22 MinT variants (eleven configurations $\times$ two covariance estimators), ranked by improvement over the baseline. In contrast to top-down disaggregation, where every configuration improved on the baseline, MinT improves on the baseline in only one of the 22 variants: MSTL Spectral $k = 3$, which reduces the system SAE by 0.55% (Sample) and 0.47% (Shrinkage). The MSTL Spectral $k = 4$ variants are essentially neutral, and all remaining variants increase the error, by up to 4.35%. The two estimators behave very similarly throughout, differing by at most a few tenths of a percentage point, and neither is consistently better.

These results reverse the top-down findings in two ways. First, where every configuration improved on the baseline under top-down, only one does so under MinT. Second, the ranking of configurations is roughly inverted: the feature-based clusters, which were best under top-down, are worst here, while the MSTL clusters, which were weakest under top-down, are the only ones MinT does not harm. This happens because top-down and MinT depend on different cluster characteristics. Top-down works best with homogeneous clusters, because a clean aggregate is easy to forecast and split. MinT instead needs useful error structure between the cluster and skill levels to exploit. When clusters are homogeneous, the skill- and cluster-level forecasts are already accurate and consistent, so there is little such structure left for MinT to use. The more mixed MSTL clusters leave slightly more of this cross-level error structure, which is why MSTL Spectral $k = 3$ is the only variant that improves.

A second reason is estimation. MinT needs the error covariance matrix W, which here is about 178×178 (roughly 30,000 entries) but is estimated from only around 700 training days. With so few days relative to its size, W is too noisy to estimate and invert reliably, so the reconciliation mostly adds noise. The Ledoit-Wolf shrinkage estimator is meant to help with exactly this, but it gives almost the same results as the sample estimator, showing that regularising W is not enough to recover useful structure that the homogeneous clusters have already removed. MinT would be expected to perform better in a different setting, with a more heterogeneous hierarchy and more days relative to the number of series, so that W can be estimated reliably.

Comparing the two reconciliation methods directly makes the difference clear. Under top-down, all eleven configurations improved on the baseline (13.63%–22.94%, Table 5.3), whereas under MinT only one did (MSTL Spectral $k = 3$, by 0.55%) and the rest increased the error,

Table 5.6: MinT reconciliation results for all eleven stable configurations under the two-level hierarchy, with the Sample and Shrinkage estimators, ranked by improvement over the baseline. The individual baseline system SAE is 781,936. Improvement is the percentage reduction in system SAE relative to the baseline (negative values indicate that reconciliation increased the error); the final column reports how many of the 175 skills improved under reconciliation.

Matrix	Method	k	Estimator	Improvement (%)	Skills improved
MSTL	Spectral	3	Sample	0.55	87
MSTL	Spectral	3	Shrinkage	0.47	79
MSTL	Spectral	4	Shrinkage	−0.03	82
MSTL	Spectral	4	Sample	−0.09	79
Features	Spectral	5	Shrinkage	−1.53	70
Features	Spectral	5	Sample	−1.59	71
Features	Hierarchical	3	Sample	−3.26	70
Features	Hierarchical	5	Sample	−3.37	71
Features	Hierarchical	7	Sample	−3.44	73
Features	Spectral	2	Shrinkage	−3.48	70
Features	Hierarchical	4	Sample	−3.50	71
Features	Spectral	2	Sample	−3.54	74
Features	Hierarchical	2	Shrinkage	−3.57	70
Features	Hierarchical	3	Shrinkage	−3.58	67
Features	Hierarchical	2	Sample	−3.70	73
Features	Hierarchical	4	Shrinkage	−3.75	66
Features	Hierarchical	5	Shrinkage	−3.75	66
Features	Hierarchical	6	Sample	−3.84	69
Features	Spectral	3	Shrinkage	−3.99	67
Features	Hierarchical	7	Shrinkage	−4.05	66
Features	Hierarchical	6	Shrinkage	−4.11	66
Features	Spectral	3	Sample	−4.35	73

by up to 4.35% (Table 5.6). Top-down beats both MinT estimators for every configuration, and the two methods rank the configurations in almost the opposite order, for the reasons discussed earlier in this section: top-down benefits from homogeneous clusters, whereas MinT needs the cross-level error structure that homogeneous clusters remove. A further point is that MinT minimises squared error while the evaluation uses absolute error, so with the spiky call-volume errors the two criteria do not always agree. Both methods are coherent, but only top-down also improves accuracy.

5.5 An Exploratory Hybrid (70/15/15)

Top-down wins overall, but MinT might still be better for a subset of skills. To test this, an exploratory hybrid used a per-skill rule: the data was re-split into 70% training, 15% validation, and 15% testing, and the validation window was used only to decide, per skill, whether top-down or MinT (Sample) had the lower SAE. This choice does not change the forecasts, so no information leaks into the test window. Because the split differs from Sections 5.3 and 5.4, the percentages below are not comparable to the 70/30 results; here the baseline system SAE on the final 15% is 533,788.

Table 5.7 shows the same pattern as before: pure top-down improves strongly (14.51%–24.57%), pure MinT is worse than the baseline everywhere (−1.27% to −6.73%), and the hybrid lands almost exactly on pure top-down. The hybrid never beats top-down by a meaningful margin.

Table 5.7: Exploratory hybrid results on the 70/15/15 split, evaluated on the final 15% test window. The baseline system SAE on this window is 533,788. Values are the percentage improvement in system SAE over the baseline; the final column reports how many of the 175 skills were assigned to top-down by the per-skill validation selection.

Matrix	Method	k	Top-Down (%)	MinT (%)	Hybrid (%)	Skills → TD
Features	Spectral	2	24.57	−6.15	23.98	96
Features	Hierarchical	2	24.44	−6.29	23.13	99
Features	Spectral	3	24.38	−6.73	24.33	102
Features	Hierarchical	3	23.88	−5.88	23.31	96
Features	Hierarchical	4	23.78	−6.21	23.23	98
Features	Hierarchical	5	23.76	−6.30	23.10	98
Features	Hierarchical	7	23.60	−6.06	23.43	97
Features	Hierarchical	6	23.54	−6.44	23.55	100
Features	Spectral	5	18.93	−3.31	20.33	101
MSTL	Spectral	4	15.29	−2.18	14.96	97
MSTL	Spectral	3	14.51	−1.27	14.92	90

The selection counts explain why: top-down was chosen for 90 to 102 of the 175 skills, a clear majority, and the skills routed to MinT gave no net gain on test. A skill that looks better under MinT on the short validation window often reverts on the test window, so the per-skill selection adds noise rather than information. The hybrid therefore keeps the strength of top-down but gains nothing from MinT.

Across both the 70/30 evaluation and the 70/15/15 hybrid, the conclusion is the same: top-down disaggregation is the more effective reconciliation method for this contact centre. MinT guarantees coherence but does not improve accuracy, and letting each skill choose between the two does not beat using top-down everywhere. The recommended pipeline is therefore feature-based clustering with top-down disaggregation, the simplest approach considered, which gives the largest and most consistent accuracy gains while staying coherent across levels.

Chapter 6

Discussion

6.1 Summary of Key Findings

Before turning to the business implications and limitations, this section recaps the main results established in Chapter 5. On the clustering side, only the feature-based and MSTL representations produced groupings that were both well separated and stable; the correlation- and shape-based measures (DCCA, Scale-Space, and DTW) and their consensus did not place a single configuration in the validity shortlist. Among the algorithms, hierarchical and spectral clustering yielded stable partitions while k-medoids did not, leaving eleven stable configurations in total, nine of them feature-based.

On the forecasting side, top-down disaggregation improved on the individual-skill baseline for all eleven configurations, reducing system-level error by between 13.6% and 22.9%, with the feature-based clusters performing best. The median per-skill improvement was positive at every volume level, but the largest absolute gains were concentrated in the high-volume skills.The benefit was most pronounced for the held-out sparse skills, whose error fell by around 90%, showing that data-driven grouping extends usable forecasts to low-coverage queues that cannot be modelled reliably on their own. The held-out new skills also improved, showing the approach benefits them as well.

Finally, the two reconciliation methods behaved very differently. Both produced forecasts that were coherent across the cluster and skill levels, but only top-down disaggregation also improved accuracy: MinT improved on the baseline in just one of the twenty-two variants tested and worsened it in nearly all others, with the two methods favouring opposite cluster structures. MinT did not work well here mainly because it relies on estimating a large error-covariance matrix (about 178×178) from only around 700 training days, which is far too few to estimate it reliably, so the reconciliation mostly added noise; the Ledoit-Wolf shrinkage estimator, meant to address exactly this, gave almost identical results. MinT would therefore be expected to perform better with fewer series, a longer training period, or both, and in a more heterogeneous hierarchy that leaves genuine cross-level error structure to exploit. The overall recommendation that emerges is feature-based clustering with top-down disaggregation, the simplest method considered and also the most accurate.

6.2 Business Implications for CCmath

The results lead to a clear recommendation for CCmath: feature-based clustering combined with top-down disaggregation. This was the most accurate pipeline tested, reducing the system-level forecast error by roughly 23% compared with the company's current per-skill baseline, and it is also the simplest, which makes it well suited for integration into the existing forecasting tool. A more accurate forecast feeds directly into better staffing decisions, reducing both the over-staffing that drives up labour cost and the under-staffing that causes service-level breaches.

A further and arguably more important benefit is the reduction in computation time. The current approach fits one model per skill, so 175 separate ETS models must be estimated. With clustering and top-down disaggregation, only the cluster-level series are forecast, between two and seven for the stable configurations in this study, and the skill-level forecasts are recovered through a simple proportional split that is computationally trivial. The expensive model-fitting step therefore drops from 175 models to a handful. For a provider that maintains forecasts for many clients, each with hundreds of skills, this lowers running costs and makes the system far easier to scale.

The approach also extends usable forecasts to skills that currently cannot be modelled reliably. The held-out sparse and new skills, too incomplete to forecast individually, were given accurate forecasts simply by assigning them to a cluster and disaggregating its forecast down to them. This is directly relevant for CCmath, where new and low-volume queues appear regularly and would otherwise be dropped or forecast poorly. Grouping skills on behaviour rather than on business labels such as department or language also replaces manual, intuition-based grouping with an automated, data-driven step. Finally, the recommended pipeline produces forecasts that are coherent across levels, so skill-level forecasts always add up to their cluster totals, removing the conflicting-numbers problem planners face when levels are forecast independently. MinT, by contrast, is not recommended here: it guarantees coherence but did not improve accuracy and was more expensive to compute. The clustering step can be re-estimated periodically, for example each quarter, so the groupings and proportions stay aligned with demand as it drifts.

6.3 Limitations

Several limitations should be kept in mind when interpreting the results of this study. First, the analysis is based on a single dataset from one of CCmath's clients. Although the dataset is large and contains a diverse set of skills, the findings may not generalise directly to other contact centres with different demand structures, and confirming the results would require testing on data from multiple clients.

Second, all modelling was performed on daily aggregated call volumes rather than the 15-minute granularity at which the data is originally recorded. Daily data preserves the dominant weekly demand pattern and was sufficient for evaluating the clustering and reconciliation methods, but staffing and scheduling decisions ultimately require intraday forecasts. Related to this, the study does not account for intraday patterns, opening hours, public holidays, or special events such as campaigns and outages. These factors strongly affect call volumes in practice, and incorporating them would be necessary before the approach is used for operational scheduling.

Third, the forecasts were evaluated on a single held-out test period covering the final 30% of the data. No rolling-origin evaluation was performed, so the reported error estimates depend on the characteristics of this particular window. The test period also falls within a strong long-term downward trend in aggregate demand, which may favour pooling-based methods that borrow strength across skills, so the size of the improvement could differ in periods with more stable or rising demand.

Fourth, around 15% of the training values were imputed, partly using an ETS-based procedure. The similarity measures, clusters, and base forecasts are therefore built in part on model-generated rather than purely observed values, especially in the earlier period where missingness was highest. While the imputation was shown to have limited effect on the aggregate demand signal, it may still influence the resulting skill groupings.

Fifth, all forecasts rely on a single base model, ETS(A,A,A). The improvement from clustering and disaggregation was not tested with other forecasting methods, so it is not known whether the same benefit would hold with, for example, ARIMA or machine-learning models. Part of the gain may be specific to the limitations of ETS at the individual-skill level. Finally, the disaggregation step has structural limits. The day-of-week proportions are static, estimated once from the training data and held fixed across the forecast horizon, so they cannot adapt to changes in the relative shares of skills within a cluster over time.

6.4 Relation to Prior Literature

The finding that MinT does not improve on top-down, and often worsens accuracy, appears to contradict the broad evidence from retail, energy, tourism, and public-health forecasting, where reconciliation consistently improves on independent and top-down approaches [1, 8, 21]. The difference lies in the hierarchy. In those applications the hierarchy is fixed and organisationally meaningful, so the levels carry genuinely different information and the lower-level series retain idiosyncratic structure that MinT can exploit. Here the hierarchy is instead built to be as homogeneous as possible, since skills are grouped by behavioural similarity. This removes much of the cross-level error structure MinT relies on: the clustering step already cleans the noisy low-volume series through pooling, so by the time reconciliation is applied the skill- and cluster-level forecasts are largely consistent and there is little left to correct. This is consistent with the literature, which reports that MinT's largest gains occur precisely at noisy, low-volume levels with inconsistent forecasts [8] — exactly the inconsistency that clustering eliminates beforehand.

Two further factors work against MinT here: the error-covariance matrix is hard to estimate reliably when many skills are reconciled over a short daily training window [30], and MinT minimises squared error while the evaluation uses absolute error on spiky, overdispersed call data, so the criterion it optimises is not the one being measured. The result is therefore consistent with the reconciliation literature rather than at odds with it: MinT helps when the hierarchy leaves exploitable structure between levels, and the homogeneous clusters used here are designed to leave very little.

Chapter 7

Conclusions

This thesis investigated whether data-driven skill-merging strategies, combined with hierarchical forecasting, can improve the accuracy and coherence of call arrival forecasts in contact centres compared with current workforce management practice. This chapter answers the research questions, summarises the main contributions, and outlines directions for future work.

7.1 Answers to the Research Questions

SQ1: Which pairwise time series similarity measure best captures the dependencies between contact center skills, considering both correlation-based and shape-based distance measures? No single time-series measure was adequate on its own. The correlation- and shape-based measures (DCCA, Scale-Space, and DTW) were all dominated by the strong day-of-week pattern shared by almost all skills, which made most skill pairs look similar and left little signal to separate them. None of these measures, nor the consensus matrix combining them, produced groupings that passed the validity and stability checks. Among the time-series measures, only MSTL gave stable groupings, because its decomposition keeps the trend and residual components that still distinguish skills once the common seasonal pattern is removed. The most informative representation, however, was the feature-based one, which describes each skill by its operational profile. This produced the clearest and most stable groupings, showing that behavioural features capture inter-skill structure better than raw similarity in this setting.

SQ2: How stable are the identified skill groups across different clustering algorithms, and does the choice of similarity measure affect the robustness of the resulting groupings? Stability depended strongly on both the algorithm and the representation. Eleven configurations passed the perturbation test, and all of them came from either the feature-based or the MSTL representation, using hierarchical or spectral clustering. No k-medoids configuration was stable, as its partitioning was sensitive to small perturbations given the very unequal cluster sizes. The feature-based representation dominated the stable set, with several hierarchical solutions completely unchanged across all perturbations. The choice of similarity measure therefore had a decisive effect on robustness: only the feature-based and MSTL representations produced stable groupings, while the correlation- and shape-based measures did not.

SQ3: Does data-driven clustering improve the forecastability of aggregated skill series, measured by forecast error, compared to the current practice of forecasting each skill individually? Yes. Under top-down disaggregation, all eleven stable configurations outperformed the individual-skill baseline, which corresponds to the partner company's current practice, reducing the system-level error by between 13.6% and 22.9%. The feature-based clusters performed best, confirming that grouping skills by behavioural similarity produces aggregates that are smoother and easier to forecast. The improvement appeared at every volume level, as the median per-skill error fell for low-, mid-, and high-volume skills alike, although the largest absolute gains came from the high-volume skills. The benefit was clearest for the held-out sparse skills, which cannot be forecast reliably on their own: assigning them to existing clusters and disaggregating reduced their error by roughly 90%, showing that data-driven grouping extends usable forecasts to low-coverage queues.

SQ4: How much does hierarchical forecast reconciliation improve the coherence and accuracy of forecasts across different aggregation levels of the contact center? Reconciliation improved coherence but not accuracy. Both top-down disaggregation and MinT produced forecasts that were coherent across the cluster and skill levels, but only top-down also improved accuracy. MinT improved on the baseline in just one of the twenty-two variants tested and worsened accuracy in nearly all others. The two methods favoured opposite cluster structures: the homogeneous feature-based clusters that were best for top-down left almost no cross-level error structure for MinT to exploit, while the more mixed MSTL clusters were the only ones MinT did not harm. For this contact centre, reconciliation therefore delivers consistency, but the accuracy gains come from the clustering and top-down disaggregation rather than from MinT.

Main research question. Taken together, the results show that data-driven skill-merging strategies, combined with top-down hierarchical forecasting, can improve both the accuracy and the coherence of call arrival forecasts compared with current practice. Grouping skills by behavioural features and forecasting at the cluster level reduced system-level error by roughly 23% while producing forecasts that are consistent across levels, which answers the central research question positively.

7.2 Summary of Contributions

This thesis combined data-driven skill clustering with hierarchical forecasting in a contact centre setting, a combination that, to the best of the author's knowledge, had not been studied before. It compared four time-series similarity measures and a feature-based representation across three clustering algorithms, screened the resulting groupings for validity and stability, and evaluated two reconciliation methods against the company's current per-skill baseline. The main findings are that behavioural feature-based clustering produces the most stable and forecastable groupings; that combining this clustering with top-down disaggregation reduces system-level forecast error by around 23% while staying coherent across levels; that the approach extends usable forecasts to sparse, low-coverage skills that cannot be modelled individually; and that

MinT reconciliation is not beneficial when the hierarchy is built from homogeneous data-driven clusters. The recommended pipeline, feature-based clustering with top-down disaggregation, is both the most accurate and the simplest method considered, and it greatly reduces computation by forecasting only a handful of cluster series instead of every individual skill.

7.3 Future Work

Several directions follow naturally from this work. The most immediate is to repeat the analysis at the 15-minute granularity required for operational staffing, and to include the intraday patterns, opening hours, public holidays, and special events that the daily analysis leaves out. Testing the approach on data from several clients would show how well the findings generalise beyond a single contact centre. On the methodological side, the static day-of-week proportions could be replaced with dynamic proportions that adapt over the forecast horizon, and the clustering benefit could be tested with base forecasters other than ETS, such as ARIMA or machine-learning models, to see how much of the improvement is specific to exponential smoothing. Finally, a rolling-origin evaluation across several test windows would give more robust estimates of forecast accuracy than the single held-out period used here.

Bibliography

[1] George Athanasopoulos, Roman A Ahmed, and Rob J Hyndman. "Hierarchical forecasts for Australian domestic tourism". In: *International Journal of Forecasting* 25.1 (2009), pp. 146–166.

[2] Kasun Bandara, Rob J Hyndman, and Christoph Bergmeir. "MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns". In: *International Journal of Operational Research* 52.1 (2025), pp. 79–98.

[3] D Manning Christopher, Raghavan Prabhakar, and Schutze Hinrich. *Introduction to information retrieval*. 2008.

[4] Joost CF De Winter, Samuel D Gosling, and Jeff Potter. "Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data." In: *Psychological methods* 21.3 (2016), p. 273.

[5] Bernard Desgraupes. "Clustering indices". In: *University of Paris Ouest-Lab Modal'X* 1.1 (2013), p. 34.

[6] Charles W Gross and Jeffrey E Sohl. "Disaggregation methods to expedite product line forecasting". In: *Journal of Forecasting* 9.3 (1990), pp. 233–254.

[7] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[8] Rob J Hyndman, Alan J Lee, and Earo Wang. "Fast computation of reconciled forecasts for hierarchical and grouped time series". In: *Computational Statistics & Data Analysis* 97 (2016), pp. 16–32.

[9] Rob J Hyndman et al. "A state space framework for automatic forecasting using exponential smoothing methods". In: *International Journal of forecasting* 18.3 (2002), pp. 439–454.

[10] Rob J Hyndman et al. "Optimal combination forecasts for hierarchical time series". In: *Computational statistics & data analysis* 55.9 (2011), pp. 2579–2589.

[11] Rouba Ibrahim and Pierre L'Ecuyer. "Forecasting call center arrivals: Fixed-effects, mixed-effects, and bivariate models". In: *Manufacturing & Service Operations Management* 15.1 (2013), pp. 72–85.

[12] Rouba Ibrahim et al. "Modeling and forecasting call center arrivals: A literature survey and a case study". In: *International Journal of Forecasting* 32.3 (2016), pp. 865–874.

[13] Boris Iglewicz and David C Hoaglin. *Volume 16: how to detect and handle outliers*. Quality Press, 1993.

[14] Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. "Fuzzy clustering of time series data using dynamic time warping distance". In: *Engineering Applications of Artificial Intelligence* 39 (2015), pp. 235–244.

[15] Abhay Jha et al. "Clustering to forecast sparse time-series data". In: *2015 IEEE 31st international conference on data engineering*. IEEE. 2015, pp. 1388–1399.

[16] Ladislav Kristoufek. "Measuring correlations between non-stationary series with DCCA coefficient". In: *Physica A: Statistical Mechanics and its Applications* 402 (2014), pp. 291–298. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2014.01.058. URL: https://www.sciencedirect.com/science/article/pii/S037843711400079X.

[17] Olivier Ledoit and Michael Wolf. "A well-conditioned estimator for large-dimensional covariance matrices". In: *Journal of Multivariate Analysis* 88.2 (2004), pp. 365–411.

[18] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 competition: Results, findings, conclusion and way forward". In: *International Journal of Forecasting* 34.4 (2018), pp. 802–808.

[19] Fionn Murtagh and Pedro Contreras. "Algorithms for hierarchical clustering: an overview, II". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7.6 (2017), e1219.

[20] Florêncio Mendes Oliveira Filho, Everaldo Freitas Guedes, and Paulo Canas Rodrigues. "Networks analysis of Brazilian climate data based on the DCCA cross-correlation coefficient". In: *Plos one* 18.9 (2023), e0290838.

[21] Anastasios Panagiotelis et al. "Forecast reconciliation: A geometric view with new insights on bias correction". In: *International Journal of Forecasting* 37.1 (2021), pp. 343–359.

[22] Leena Pasanen and Lasse Holmström. "Scale space multiresolution correlation analysis for time series data". In: *Computational Statistics* 32.1 (2017), pp. 197–218.

[23] Lin Piao and Zuntao Fu. "Quantifying distinct associations on different temporal scales: comparison of DCCA and Pearson methods". In: *Scientific Reports* 6.1 (2016), p. 36759.

[24] Paulo E Rauber, Alexandre X Falcao, Alexandru C Telea, et al. "Visualizing Time-Dependent Data Using Dynamic t-SNE." In: (2016).

[25] Jorge M Santos and Mark Embrechts. "On the use of the adjusted rand index as a metric for evaluating supervised classification". In: *International conference on artificial neural networks*. Springer. 2009, pp. 175–184.

[26] Erich Schubert and Peter J Rousseeuw. "Fast and eager k-medoids clustering: O (k) runtime improvement of the PAM, CLARA, and CLARANS algorithms". In: *Information Systems* 101 (2021), p. 101804.

[27] Mahya Seyedan, Fereshteh Mafakheri, and Chun Wang. "Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach". In: *Decision Analytics Journal* 3 (2022), p. 100033.

[28] James W Taylor. “Density forecasting of intraday call center arrivals using models based on exponential smoothing”. In: *Management Science* 58.3 (2012), pp. 534–549.

[29] Ulrike Von Luxburg. “A tutorial on spectral clustering”. In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[30] Shanika L Wickramasuriya, George Athanasopoulos, and Rob J Hyndman. “Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization”. In: *Journal of the American Statistical Association* 114.526 (2019), pp. 804–819.

[31] GF Zebende, AA Brito, and AP Castro. “DCCA cross-correlation analysis in time-series with removed parts”. In: *Physica A: Statistical Mechanics and its Applications* 545 (2020), p. 123472.