Forecasting and evaluating intermittent demand with timing-aware global models and heterogeneous data

Jonatan Flyckt[a,b,*], Florian Westphal[c], Niklas Lavesson[b]

[a]Herenco AB, Skolgatan 24, Jönköping, 553 16, Sweden
[b]Blekinge Institute of Technology, Valhallavägen 10, Karlskrona, 371 79, Sweden
[c]Jönköping University, Gjuterigatan 5, Jönköping, 553 18, Sweden

Abstract

Intermittent demand arises from sporadic purchasing, resulting in time-series with many zero values. Traditional demand rate forecasting methods provide no information about when demands will occur, limiting their usefulness. We propose a global forecasting approach that decomposes intermittent time-series into demand size and interval length components, reduces data skewness through pre-processing, and trains a deep learning model to predict both timing and magnitude. To evaluate such forecasts, we introduce Intermittent Alignment Error (IAE), which captures timing and magnitude performance across heterogeneous datasets. Our approach outperforms demand rate and non-parametric baselines by 25–40%, improving both timing performance and total percentage error while requiring shorter context and generalising across domains. The study demonstrates the feasibility of structured point forecasts and introduces a robust evaluation metric for intermittent demand, both of which have the potential to improve decision making in domains such as make-to-order manufacturing with mass customisation, where demand timing is critical.

Keywords: Demand forecasting, Intermittent demand, Error measures, Model selection, Evaluating forecasts, Neural networks, Time series,

[⋆]This research was supported by the KKS foundation through the S.E.R.T. Research Profile project www.rethought.se at Blekinge Institute of Technology).

[*]Corresponding author

Email addresses: jonatan.flyckt@bth.se (Jonatan Flyckt), florian.westphal@ju.se (Florian Westphal), niklas.lavesson@bth.se (Niklas Lavesson)

1. Introduction

Intermittent demand arises in domains such as spare parts, electronics retail (Kourentzes, 2013) and make-to-order manufacturers with mass customisation. It is characterised by sporadic demand interspersed with zero-demand periods. These patterns are difficult to forecast using classical methods like ARIMA (Box and Jenkins, 1970) which assume more continuous demand. Classical intermittent time-series models such as Croston (Croston, 1972) and its derivatives (Teunter et al., 2011; Shale et al., 2006) produce demand rate forecasts, i.e., forecasts which estimate a constant rate of demand across a forecasting horizon, unlike point forecasts, which predict an expected demand value for each time-step. Demand rate forecasts are useful to determine reorder points in stock-keeping situations, but insufficient in settings like make-to-order manufacturing with mass customisation, where thousands of unique articles are produced only after orders are placed.

Although make-to-order companies wish to minimise inventory due to obsolescence risk, they may still need to proactively purchase or produce components to meet tight deadlines. For such occasions, demand forecasts need both accurate timings and magnitudes to minimise the risk of mistimed decisions, especially if the demand is highly intermittent, i.e., the average time-periods between demands are long. Deep learning models can be used to forecast demands (Salinas et al., 2020; Türkmen et al., 2021; Jeon and Seong, 2022). However, they will often have a bias towards under-forecasting or only predicting zero demands and perform better on less skewed distributions (Yang et al., 2021).

We present a deep learning approach which uses pre-processing to scale and undersample demands and decompose each demand occurrence into its zero-demand interval length and demand size. Training separate local models for thousands of unique time-series is impractical. Instead, our approach uses a global transformer model trained across all time-series. The model predicts *one-demand-ahead*, generating a full forecasting horizon by recursively feeding predictions back into the model. This produces point forecasts of the expected demand in the same unit as the unprocessed time-series. Using the last n demand occurrences rather than a fixed time window allows a single global model to forecast demands across multiple time-series with large variations in intermittency and demand magnitudes. We are specifically interested in severe intermittency but also evaluate the model in scenarios with varying context lengths and intermittency levels using a retail dataset from the M5 competition (Makridakis et al., 2022). Additionally, we evaluate how well the global model can generalise to a separate dataset from a make-to-order packaging manufacturer.

Because most metrics for intermittent demand are aimed at demand rate forecasts (Kourentzes, 2014; Martin et al., 2020; Wallström and Segerstedt, 2010), and per-time-step error metrics such as Mean Squared Error (MSE) and Mean Absolute Error (MAE) would bias in favour of zero-demand forecasts (Kourentzes, 2014; Wallström and Segerstedt, 2010), we introduce a novel metric that accounts for both demand timing and magnitude. The purpose of the metric is to aid in model selection. Because the global model handles time-series with varying intermittency and demand magnitudes, the metric is designed to be agnostic to different characteristics: it requires good timing for low intermittency but is more lenient for highly intermittent time-series. We evaluate the fairness and robustness of the metric in an experiment by using synthesised forecasts with increasing timing and magnitude errors across varying intermittency levels.

The study aims to answer the following research questions:

RQ1 : How well can global deep learning-based demand forecasting models generalise to datasets of heterogeneous intermittent time-series to predict both demand timing and magnitudes?

RQ2 : How well can one metric measure the correctness of forecasts, taking both demand timing and magnitude into account, for heterogeneous collections of intermittent time-series?

2. Related work

Demand forecasting supports decisions by estimating the future values of time-series as probabilistic distributions or point estimates, either per time-step or across an entire forecasting horizon. Per time-step point estimates can be generated recursively by predicting one step at a time and feeding predictions back into the model, or directly, by forecasting all future time-steps simultaneously. Intermittent demand is commonly defined as having a high Average Demand Interval (ADI) and a low demand variability (CV^2), with the standard thresholds being $ADI \geq 1.32$ and $CV^2 < 0.49$ (Syntetos et al., 2005).

Most intermittent demand forecasting methods estimate the demand rate across a forecasting horizon rather than the demand occurrences and magnitudes at specific time-steps. Classical approaches such as the Croston method (Croston, 1972) and its derivatives TSB (Teunter et al., 2011) and SBJ (Shale et al., 2006) produce fixed-rate forecasts. Such forecasts are effective in stock-keeping domains but insufficient for domains such as make-to-order manufacturing, where both demand timing and magnitude are important. Non-Parametric Time-Series (NPTS) models can produce forecasts which measure both timing and demand, but they lack generalisability as they essentially sample from the input to create the output (Han et al., 2017; Alexandrov et al., 2020).

Many deep learning models have been proposed to forecast intermittent demand, mainly using sequential models such as LSTMs (Hochreiter and Schmidhuber, 1997) or transformers (Vaswani et al., 2017), but non-sequential models have seen successful use as well (Kourentzes, 2013). Deep learning models for time-series forecasting are typically trained either locally, on a single time-series, or globally, across many series to learn shared patterns (Salinas et al., 2020). Whether models are trained locally or globally is not always detailed in published works, despite its growing importance in demand forecasting. Recent research has explored the effects of global models, and whether it is possible to construct pre-trained foundation models capable of forecasting time-series from completely new domains (Goswami et al., 2024; Liang et al., 2024). Another area of growing interest is the impact of training set heterogeneity on global model performance, and whether semi-global models trained on homogeneous groups of time-series yield better results (Bandara et al., 2020; Oriona et al., 2023; Abbasimehr and Noshad, 2025; Sonnleitner, 2025).

The over-representation of zeros among target values in raw intermittent demands causes issues for deep learning models, which often default to under-predicting demands or only predicting zeros (Yang et al., 2021). Some deep learning methods bypass this issue, such as DeepAR which uses a negative binomial distribution and fits the model to this distribution at each time-step to capture uncertainty through probabilistic forecasts that vary over time (Salinas et al., 2020). Another example is Deep Renewal Processes (DRP) (Türkmen et al., 2021), which models the demand as a renewal process using decomposed inter-demand interval lengths and magnitudes. In practice, DeepAR's and DRP's outputs resemble demand rate forecasts, but contain elements of variation and uncertainty. Jeon and Seong (2022) proposed a Tweedie-distribution-based modification to DeepAR to address the challenge of zero-inflated targets in intermittent demand forecasting. Although the final output is a point forecast derived by averaging samples, the model itself is trained using the Tweedie distribution. Many deep learning approaches decompose the time-series into demand and interval length components (Kourentzes, 2013; Rožanec et al., 2022) similar to Croston (Croston, 1972). We use a similar decomposition approach but present a new way to pre-process the time-series to work well with global deep learning models for point forecasts. To our knowledge, few deep learning approaches have focused on generating structured point forecasts that capture both demand timing and magnitude across the forecasting horizon, rather than demand rates or probabilistic distributions, particularly in a global modelling context.

Effective model selection depends on error metrics that match the nature of the forecasting task. Common per-time-step metrics like MSE, MAE, and Symmetric Mean Absolute Percentage Error (sMAPE) are poorly suited for intermittent demand; the high prevalence of zeros introduces a bias toward under-forecasting or predicting zeros (Martin et al., 2020; Wallström and Segerstedt, 2010; Kourentzes, 2014). Several alternative metrics have been proposed to address this, including Stock-keeping-oriented Prediction Error Cost (SPEC) (Martin et al., 2020) and Periods In Stock (PIS) (Wallström and Segerstedt, 2010), which aim to simulate how well forecasts contribute to accurate stock-keeping. However, these metrics are not applicable in domains that do not maintain inventory, such as make-to-order manufacturing. Mean Squared Rate (MSR) provides a domain agnostic measure of intermittent forecasts by computing rolling cumulative errors across the forecasting horizon (Kourentzes, 2014). Although this is a useful metric for local demand rate forecasts, it does not assess the timing and magnitude of demand events and performs poorly when comparing across heterogeneous time-series in a global modelling context. We therefore propose a metric that assesses both demand timing and magnitude for point forecasts across the full forecasting horizon.

3. Method

In this section, we describe the approach for the intermittent forecasting model and the data pre-processing steps taken to train it. The approach is aimed at univariate data, meaning the future values of one variable (customer demand) are forecasted by observing past values of the same variable. We also describe the proposed metric for forecasting performance which assesses both the timing and magnitude of demands. Lastly, we describe a series of experiments that aimed to test the forecasting model in different scenarios (RQ1), and to test the fairness and robustness of the proposed metric under different conditions (RQ2). We focused on time-series with large intermittency (ADI > 5) but also tested both lower and higher levels of intermittency (Section 3.4.3).

3.1. Datasets

We used the retail dataset from the M5 forecasting competition (Makridakis et al., 2022) to develop and validate our approach, both for the forecasting model and metric. The dataset is common in forecasting studies (Jeon and Seong, 2022; Türkmen et al., 2021; Kiefer et al., 2021), and consists of intermittent, lumpy, erratic, and smooth time-series. We applied the filtering criteria suggested by Syntetos et al. (2005): $ADI \geq 1.32$ and $CV^2 < 0.49$, to only include intermittent time-series. The full time-series decomposition and pre-processing approach is detailed in Section 3.2.1. Each day was treated as one time-step, and each product and location were treated as a separate time-series. We removed time-series with anomalous or missing data by identifying unusually large gaps that were at least 3 times larger than the 85th percentile of interval lengths for those series and split each series into segments at those locations.

To best mimic a real-world scenario, we split time-series into train, test, and validation parts by extracting test and validation samples from the tail end of time-series for the time-series that were long enough to support it. We first discarded time-series that did not have at least 30 demand occurrences and then extracted the last 30 demand occurrences from time-series with at least 60 demand occurrences for the test and validation portions. This allowed us to slide a window across the entire series to generate a sufficient number of samples for different context lengths. Depending on the chosen context length, roughly 9 % of samples went to the test and validation sets each, and roughly 82 % went to the training set.

We also used an additional dataset from a packaging manufacturer employing a make-to-order manufacturing strategy with mass customisation offers. This dataset helped to validate the generalisability of the approach across domains via an experiment detailed in Section 3.4.5. This dataset used each month as a time-step instead of each day because that level of forecasting granularity is sufficient for decision-making, whereas a per-day approach would have resulted in excessively intermittent series. Table 1 shows statistics for both datasets after the described data selection steps.

Dataset	**Number of time-series**		**ADI Percentiles**					**CV^2 Percentiles**				
	Total	**Segments**	**0**	**25%**	**50%**	**75%**	**100%**	**0**	**25%**	**50%**	**75%**	**100%**
M5	99 793	65 384	1.32	1.82	2.63	4.39	66.64	0.00	0.17	0.26	0.35	0.49
Packaging manufacturing	2 096	2 096	1.32	1.52	1.95	2.71	12.00	0.00	0.16	0.23	0.33	0.49

Table 1: Summary statistics of the datasets used after filtering to only include intermittent time-series of sufficient length for the study.

3.2. Modelling approach

3.2.1. Data pre-processing

The high prevalence of zeros in intermittent time-series leads to a skewed data distribution (Figure 1), making the time-series challenging for deep learning models to interpret (Yang et al., 2021). We employed several data pre-processing steps to combat this skewness. The first step was to decompose the time-series into their demand size and interval length components, where the interval length represents the number of time-steps without demand leading up to the demand occurrence. This approach has been done before by the Croston method and its derivatives (Croston, 1972; Teunter et al., 2011; Shale et al., 2006), as well as in deep learning approaches (Kourentzes, 2013; Türkmen et al., 2021). This shift reframed the forecasting task from one-step-ahead to *one-demand-ahead*, and the input from a fixed context length of time-steps to a fixed context length of demand occurrences, but with varying numbers of time-steps.

After decomposition, strong skewness was still present in both the interval length and demand size components (Figure 2a and Figure 3a), and a few common values were overrepresented. To prevent the model from simply learning to predict the most common value, we performed a time-series centric scaling to produce a greater variety of values. Each time-series s received a scaling multiplier α_s, which we defined as the multiplicative inverse of the time-series' mean multiplied by a random noise component:

$$\alpha_s = \left(\frac{\sum_{i=1}^{N_s} x_i}{N_s} \cdot \epsilon \right)^{-1} \tag{1}$$

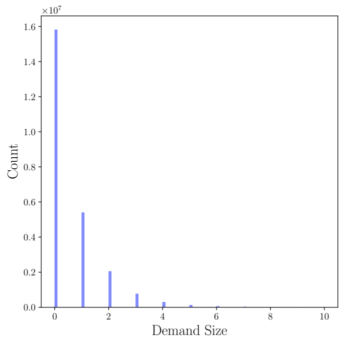


Figure 1: Histogram of raw (before decomposition) demand values across all intermittent time-series in the M5 dataset, including zeros. The high frequency of zeros (65 %) and the long-tailed distribution of non-zero values motivate the decomposition into demand sizes and interval lengths.

where:

- x_i is the original value (either demand size or interval length) at time step i,
- N_s is the number of time-steps in series s,
- $\epsilon \sim \mathcal{U}(1 - \lambda, 1 + \lambda)$ is a noise term drawn from a uniform distribution to introduce variability in the scaling, with λ set to 0.5.

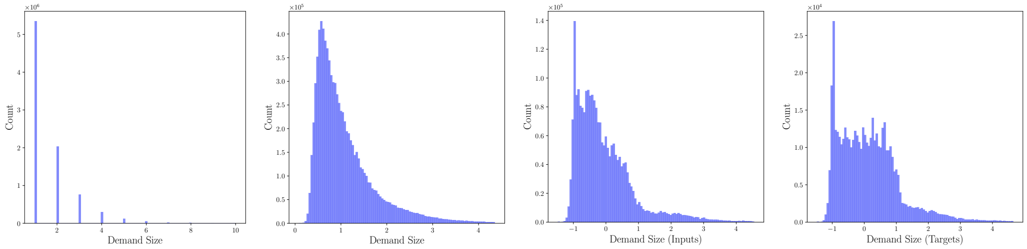
We computed the scaling multiplier based only on the lower half of the distribution for the interval length components to mitigate the effect of extreme outliers. For demand sizes, where such extreme values were rare, we used the full distribution. The scaled values $\tilde{x}_s$ for time-series s were then computed as:

$$\tilde{x}_s = x_s \cdot \alpha_s \tag{2}$$

After these scaling steps, the dataset-wide data distribution was still slightly skewed, but with more unique values than before and more diversity between the time-series (Figure 2b and Figure 3b). Importantly, for each individual time-series, the internal relations between the demands were preserved because the same scaling multiplier was applied to each value in the same series.

The final pre-processing step was applied per sample after extracting training, validation, and test splits. Because the demand size distribution remained skewed, we applied undersampling to the target values. Outliers, defined using configurable percentile thresholds, were set aside, and the remaining targets were divided into n equal-width buckets. Larger buckets were undersampled such that the largest contained at most k times as many samples as the smallest. After undersampling, we reintroduced the outliers. The number of buckets, leniency, thresholds, and whether to retain outliers were all treated as hyperparameters and tuned as described in Section 3.2.2. Lastly, we applied standard scaling to generate the final input (Figure 2c) and target (Figure 2d) values.

To avoid excessively reducing the dataset size, we applied a quantile transformation to the interval length values instead of undersampling them. This transformation maps values to a uniform $[0, 1]$ distribution using the cumulative distribution function and its inverse, ensuring an even spread and emphasising short to moderate intervals while reducing prediction errors from outliers. We then applied standard scaling to match the scale of the demand sizes, producing the final input (Figure 3c) and target (Figure 3d) values. All scalers were fitted on the training data and applied to the validation and test sets.



(a) Raw values

(b) After scaling

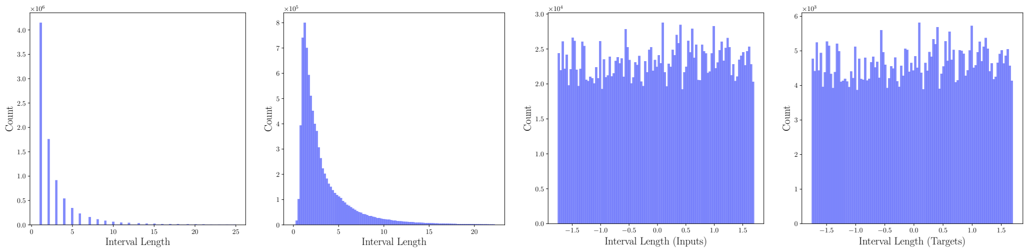
(c) Model inputs

(d) Model targets (undersampled)

Figure 2: **Demand size** data distribution during the pre-processing steps for the M5 intermittent training samples: *(a)* shows the raw demand values after decomposition, and *(b)* shows the same values after applying the series-wise scaling defined in Equation 1 and Equation 2. *(c)* shows the final input values used during training, and *(d)* shows the undersampled target values.

3.2.2. Deep learning model

We used a transformer encoder model (Vaswani et al., 2017) for the forecasting task. Several models were tested, including LSTMs (Hochreiter and Schmidhuber, 1997), 1D convolutional neural networks (1DCNN), and multilayer perceptrons (MLP). The model choice had a moderate effect on



(a) Raw values

(b) After scaling

(c) Model inputs

(d) Model targets

Figure 3: **Interval length** data distribution during the pre-processing steps for the M5 intermittent training samples: *(a)* shows the raw interval length values after decomposition, and *(b)* shows the same values after applying the series-wise scaling defined in Equation 1 and Equation 2. *(c)* and *(d)* show the final input and target values used during training after applying a quantile transformer.

forecasting performance. The sequential approaches (transformers, LSTMs, 1DCNNs) performed slightly better than the non-sequential approaches, and the transformer showed more improvements the larger the dataset was, which is one of its known strengths (Vaswani et al., 2017).

The architecture included both a positional embedding and positional encoding, followed by a configurable stack of encoder layers. Key parameters such as the number of encoder layers, attention heads, model and feed-forward dimensions, batch size, optimiser, learning rate schedule, and data loader hyperparameters were all tuned using random search with our proposed metric (Section 3.3) as the objective function. We ran the random search in two stages: first to identify a promising region of the hyperparameter space, and second to refine the search within that region. Full details are provided in Table A.8.

For all experiments, the model was trained until no improvements had been made on the validation set loss for 75 epochs (choosing the model with the lowest validation loss). Figure B.13 in the appendices shows the loss curve for the main forecasting experiment (detailed in Section 3.4.2), indicating a stable model training procedure.

3.3. Metric approach

Error metrics should reflect the nature of the forecasting task and goal to adequately support model selection and guide model development. However, it is difficult to assess intermittent time-series forecasts due to the prevalence of zeros. Traditional per-time-step metrics such as MSE and MAE over-penalise slightly mistimed forecasts (Martin et al., 2020; Wallström and Segerstedt, 2010; Kourentzes, 2014), and demand rate metrics such as MSR (Kourentzes, 2014), SPEC (Martin et al., 2020), and PIS (Wallström and Segerstedt, 2010) ignore whether the forecasted demands align time-wise with actual demand events. It is difficult to assess the timing of a forecast compared to a ground truth across a forecasting horizon, because there can be multiple demand occurrences and one would first need to determine which demands to compare to each other. To overcome this issue, we propose a new metric, *Intermittent Alignment Error (IAE)*, which is designed to:

- Evaluate both demand timing and magnitude with a single metric
- Be lenient towards mistimed forecasts proportionally to how intermittent the time-series is
- Be applicable to point forecasts across a horizon both for local models as well as global models on many heterogeneous time-series, i.e., time-series with different levels of intermittency and demand magnitudes.

The metric evaluates forecasts from two perspectives:

1. **Recall error** - how well the ground truth demands are captured by the forecast.
2. **Precision error** - how well the forecasted demands correspond to ground truth demands.

For both recall and precision error, we apply a set of expanding masks centred on demand events, and weight them to provide some leniency in the timing of the demands. Time-series with a higher intermittency (having longer average demand intervals between demand occurrences) use more masks which cover a longer time-period around demand events. The ADI for a time-series is defined as:

$$\text{ADI} = \frac{N}{n_z} \tag{3}$$

where N is the total number of time steps and n_z is the number of non-zero demand occurrences. We use the contraharmonic mean of the precision and recall errors to produce the final metric score, penalising models which only perform well in either of the two.

The *recall error* and *precision error* are computed in essentially the same manner but recall error measures how well the forecast aligns to the ground truth, and precision error measures how well the ground truth aligns to the forecast. For the recall error, we construct a set of symmetric masks around each ground truth demand at time index t. Each additional mask expands symmetrically around t, covering time-steps from $t - i$ to $t + i$. The number of masks M around each demand occurrence is determined by the historical average demand interval ADI for the ground truth of that time-series as:

$$M = \max\left(1, \left\lfloor \sqrt{\mathrm{ADI}} + 0.5 \right\rfloor\right) \tag{4}$$

Each mask m_i includes the time steps in the range $[t - i, t + i]$ (within the forecast window). For each mask m_i, an error term e_i is computed, comparing the total forecasted and actual demand values within that mask:

$$e_i = \frac{(Y_i - \hat{Y}_i)^2}{Y_i^2} \tag{5}$$

where

$$Y_i = \sum_{j=t-i}^{t+i} y_j, \quad \hat{Y}_i = \sum_{j=t-i}^{t+i} \hat{y}_j \tag{6}$$

represent the total ground truth and forecasted demand values within the mask, respectively.

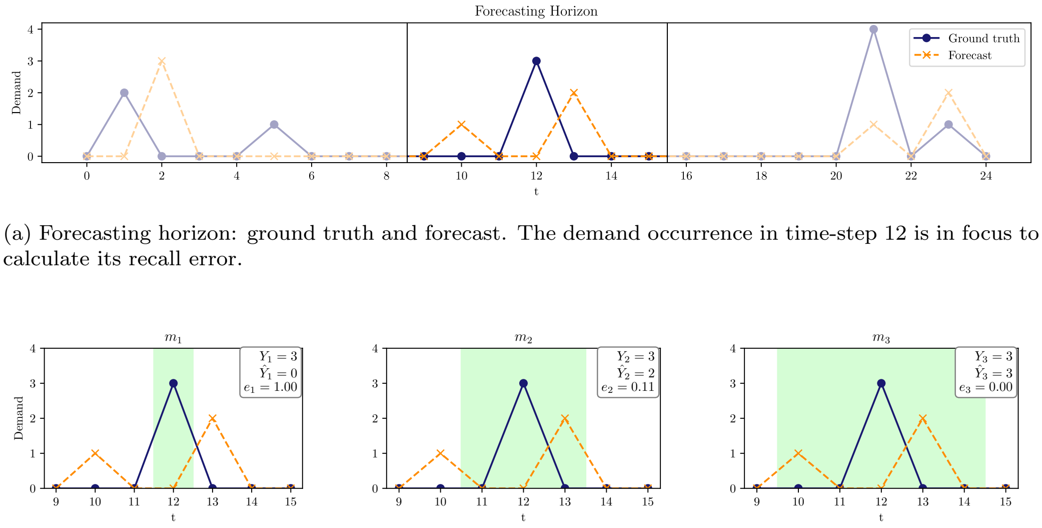
Each mask is assigned a weight w_i which is larger for the wider masks to be lenient on slightly mistimed forecasts. The weighted sum produces the error ϵ_t for that demand occurrence:

$$\epsilon_t = \sum_{i=1}^{M} w_i \cdot e_i \tag{7}$$

where w_i increases linearly the wider the mask is, and the weights are normalised to sum to 1:

$$w_i = \frac{2i}{M(M+1)}, \quad i = 1, \ldots, M \tag{8}$$

Figure 4 visualises how the metric treats a single demand occurrence to calculate the recall error, and how the expanding masks and their weights contribute to the error score.



(a) Forecasting horizon: ground truth and forecast. The demand occurrence in time-step 12 is in focus to calculate its recall error.

(b) First mask, width=1.

(c) Second mask, width=3.

(d) Third mask, width=5.

m_i	Y_i	$\hat{Y}_i$	e_i	w_i	$w_i \cdot e_i$
m_1	3	0	1.000	0.166	0.166
m_2	3	2	0.111	0.333	0.037
m_3	3	3	0.000	0.500	0.000

(e) Error and weight contributions for each mask.

$$\begin{aligned}\epsilon_t &= \sum_{i=1}^{M} w_i \cdot e_i \\ &= 0.166 + 0.037 + 0.000 \\ &= 0.203\end{aligned}$$

(f) Error aggregation equation.

Figure 4: Illustration of the recall error calculation for a single demand occurrence (at t = 12) in the IAE metric using expanding masks. Even though the forecasts are slightly mistimed, the error is low because the metric allows some leniency in the demand timing.

The total alignment error $\mathcal{E}$ is the sum of the errors across all ground truth demand occurrences where each individual error ϵ_t is weighted by the size of its corresponding ground truth demand relative to the total demand in the series. To keep the final error bounded and to dampen the influence of extreme values, we applied a bounded logistic transformation to produce the final *recall error* $\mathcal{E}_R$:

$$\mathcal{E}_R = \frac{1}{1 + \exp\left(-k \cdot (\mathcal{E} - x_0)\right)}, \quad \mathcal{E} = \sum_{t=1}^{T} \frac{y_t}{\sum_{t'=1}^{T} y_{t'}} \cdot \epsilon_t \tag{9}$$

This shifts the inflection point to x_0 and scales the curve's steepness by k, where $k = 5$ and $x_0 = 0.75$ were chosen empirically to produce values that neither saturate too early nor become overly lenient.

The *precision error* $\mathcal{E}_P$ is computed using the same procedure but focuses on how well the ground truth aligns to the forecast instead of how well the forecast aligns to the ground truth, by swapping places of the forecast and ground truth values for all calculations.

Finally, the *IAE* is computed as the contraharmonic mean of the recall and precision errors to ensure a balanced evaluation that penalises both missed demand occurrences (poor recall) and excess forecasted demand (poor precision):

$$\text{IAE}(\mathcal{E}_R, \mathcal{E}_P) = \begin{cases} \dfrac{{\mathcal{E}_R}^2 + {\mathcal{E}_P}^2}{\mathcal{E}_R + \mathcal{E}_P}, & \text{if } \mathcal{E}_R + \mathcal{E}_P \neq 0 \\ 0, & \text{if } \mathcal{E}_R + \mathcal{E}_P = 0 \end{cases} \tag{10}$$

3.4. Experiments

3.4.1. Metric experiment

To address RQ2, we performed an experiment which aimed to test the fairness, robustness, and characteristics of the IAE metric. To test the metric in different scenarios, we extracted 6 sub-datasets with different ranges of ADI from the M5 dataset (Table 1). The limits for the sub-datasets were: $[1.32, 2)$, $[2, 4)$, $[4, 7)$, $[7, 11)$, $[11, 16)$, $[16, 30)$, with 500 time-series, 100 time-steps long randomly sampled from each subset. From these ground truth time-series, we synthesised forecasts with increasing levels of error in demand timing and magnitude. We simulated timing errors by shifting demand events n steps left or right, where $n \in \{0,\ 1,\ 2,\ 3,\ 5,\ 10,\ 20\}$. Magnitude errors were introduced by perturbing each demand value y_t by a signed proportion of itself. For each time-series, a sign $s \in \{-1, 1\}$ was drawn randomly and a magnitude factor $x \in \{0,\ 0.25,\ 0.5,\ 0.75,\ 1,\ 1.5,\ 2.5\}$ was applied across that series. The perturbed demand values $\hat{y}_t$ were then computed as:

$$\hat{y}_t = y_t + s \cdot x \cdot y_t \tag{11}$$

This allowed us to study how the metric behaves as errors get worse, and across datasets with heterogeneous time-series. We visualise the resulting 2D grid of timing × magnitude errors in Section 4.1.

We performed the same experiment using a per-time-step error metric: Symmetric Mean Absolute Percentage Error (sMAPE), and a modified version of the demand rate metric MSR (Kourentzes, 2014). We adapted MSR into a new metric, Weighted Mean Squared Rate (WMSR), by evaluating cumulative forecasts instead of per-time-step forecasts, thereby aligning the units of forecast and ground truth. To enable global use of the metric, we weighted each time-series by its mean demand size (ensuring invariance to scale) and computed the mean rather than the sum of errors (ensuring invariance to forecasting horizon length). Finally, we developed a complementary metric, Sum Aggregate Percentage Error (SAPE), which measures the percentage error between the total forecasted and actual demand over the forecasting horizon H.

$$\text{SAPE} = \begin{cases} \min\left(\dfrac{\left|\sum_{t=1}^{H}\hat{y}_t - \sum_{t=1}^{H} y_t\right|}{\sum_{t=1}^{H} y_t},\ 10\right) & \text{if } \sum_{t=1}^{H} y_t > 0 \\ \text{undefined} & \text{if} \sum_{t=1}^{H} y_t = 0 \end{cases} \tag{12}$$

SAPE does not assess timing in any way but can be useful as a sanity check to determine if models find roughly correct demand sums. We capped the error at 10 to avoid exploding values. The purpose of testing the characteristics of WMSR, sMAPE, and SAPE was to highlight that current metrics do not do the following:

1. Correctly assess both the timing and magnitude error of forecasts for intermittent demand.
2. Work in a global context across heterogeneous time-series with different demand magnitudes and inter-demand interval lengths.

3.4.2. Main forecasting experiment

We conducted a series of experiments to test the model approach described in Section 3.2.2, addressing RQ1 across different scenarios. We compared it with a statistical demand rate forecasting model (TSB) (Teunter et al., 2011) as well as NPTS (Han et al., 2017; Alexandrov et al., 2020) to also compare against a statistical model which produces point forecasts with both demand timing and magnitude. Unlike our model, which uses a decomposed input structure and was trained on historical data, TSB and NPTS operate directly on the raw time-series and are not trained. Instead, they are applied directly to the test input sequences to generate forecasts for the specified horizon. We used the TSB implementation from the pyInterDemand Python library (Pereira, 2021) and the NPTS implementation from GluonTS (Alexandrov et al., 2020). The TSB parameters were set to $\mathrm{alpha} = 0.1, \mathrm{beta} = 0.1, \mathrm{n\_steps} = 1$, and for NPTS we used 100 samples and the 0.75 quantile. These parameters were selected through iterative testing guided by the IAE metric. Forecasts were evaluated using the proposed IAE and SAPE metrics in all experiments. The hyperparameters used in our model are listed in Table A.8. To generate multi-step forecasts, our model recursively fed each predicted output back into the model until the entire forecasting horizon was filled.

The main forecasting experiment was designed to evaluate the model in a general scenario with highly intermittent time-series. We focused on high-intermittency cases because existing models often perform poorly as intermittency increases, whereas for low-intermittency series, demand rate models can suffice as decision support. We selected time-series with $ADI > 5$, yielding 18 370 series, of which 13 403 (682 306 samples) were used for training, 2 483 (75 962 samples) for validation, and 2 484 (76 017 samples) for testing. Test and validation sets were selected at random. The forecasting horizon was set to 14 days (2 weeks), reflecting a common use case where forecasts are needed for a fixed-length future period to support decision-making. The input length was set to 5 demand steps. Due to variation in interval lengths, the actual number of time-steps in the input sequences varied. We also tested alternative input lengths in the experiment described in Section 3.4.4.

To better understand our model, we conducted an exploratory analysis of feature importance using a decision tree regression model trained to predict the IAE value from a set of simple time-series features, e.g., *number of future* *demands, mean interval length, ratio of future demands to input demands, input time-step length, interval length standard deviation,* and *future demand sum.* Our aim was to identify the features or combinations of features that best explained poor or good forecasting performance.

3.4.3. Varying ADI experiment

To test whether different levels of intermittency affect forecasting performance (relating to RQ1), we conducted an experiment where we varied the lowest level of intermittency in the dataset from very low: $ADI > 1.32$, to very high: $ADI > 20$, as well as three additional thresholds in between: 3, 7, and 12. This was done to understand at which cutoff the model became more useful compared to the statistical methods according to the IAE metric, and to highlight that the model is robust to different scenarios. Additionally, we conducted an experiment to test whether homogeneity in the training dataset was important for our approach. For this experiment, we evaluated the forecasts for the test dataset where $ADI > 20$ and compared the performance on those samples between the model trained on $ADI > 1.32$ and the model trained on $ADI > 20$. We used a forecasting horizon of 14 time-steps and an input length of 5 demand steps.

3.4.4. Varying context length experiment

Our experience had shown that statistical models often performed better with a longer context length (number of demand steps in the input), and we therefore wanted to test the model against TSB and NPTS with a varying context length. Because new articles can arrive frequently in a mass customisation context, and swaps to new material versions for the same product are not always tracked (causing historical product demand to be missing), it is essential to be able to forecast early in the product lifecycle. To further investigate RQ1 and test whether our model is robust to shorter (or longer) contexts, we tested context lengths of 3, 5, 10, 15, and 20. We used the same parameters as in the main forecasting experiment (Section 3.4.2): $ADI > 5$ as the selection and 14 time-steps in the forecasting horizon.

3.4.5. MTO packaging manufacturing experiment

Because the modelling approach was developed entirely on a general retail dataset, we also examined how it translated to a domain with slightly different characteristics (addressing RQ1). We conducted an experiment on a dataset from an MTO packaging manufacturer, which included many articles with intermittent demand. The aim was to explore: *How should global models for intermittent demand forecasting be applied to organisational datasets with domain-specific characteristics, such as those found in MTO settings?*

Compared to the M5 data, this dataset was aggregated monthly and had slightly lower average intermittency, though demand variability was similar (Table 1). It also contained far fewer samples: 2 096 intermittent time-series versus 65 384 usable ones in M5. Given these characteristics, we included all intermittent time-series ($ADI > 1.32$) in the experiment. We used a context length of 5 demands to forecast 6 time-steps (months) ahead, testing on 152 randomly selected time-series (3 800 samples), with 32 799 training and 3 800 validation samples. The model's performance on the test dataset was evaluated using the following approaches:

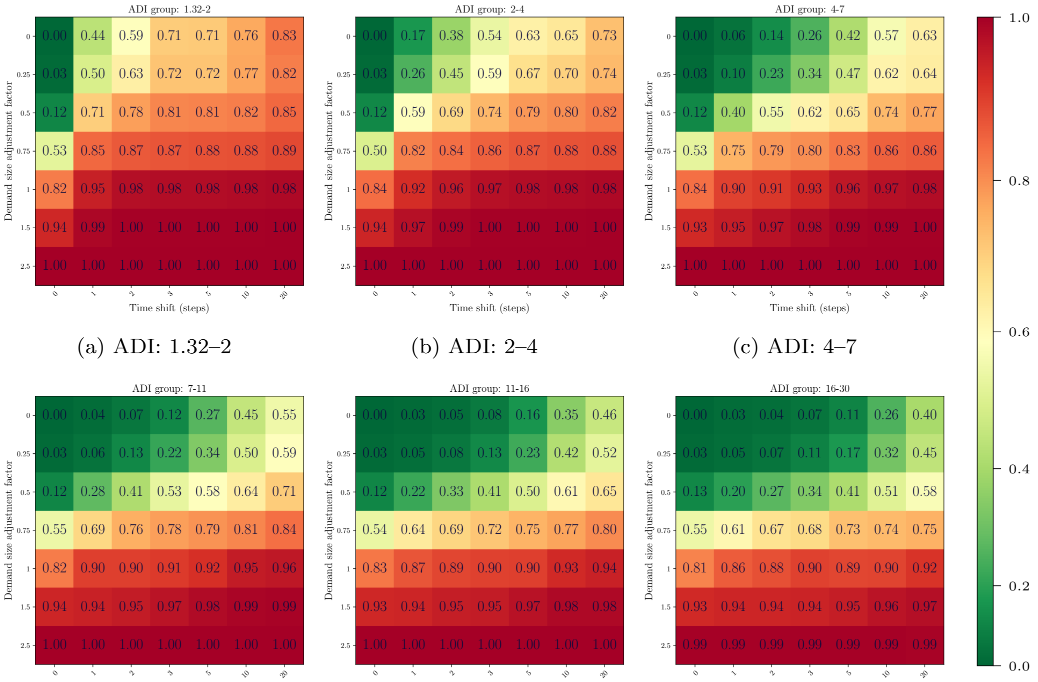
1. Training only on the M5 dataset.
2. Training only on the packaging dataset.
3. Training on both the M5 dataset and the packaging dataset together.
4. Pre-training on the M5 dataset and fine-tuning on the packaging dataset.

4. Results and analysis

4.1. Metric analysis

This section analyses the results of the Metric Experiment (Section 3.4.1) by displaying them as 2-dimensional matrices. Figure 5 shows the result matrices for our proposed metric IAE. Each matrix shows the metric scores for one dataset, and each cell shows the score for a time and demand shift (synthetically derived from its ground truth). As the ADI increases, the metric is more lenient on mistimed forecasts, but stays consistent in the assessment of demand magnitudes. For example: With an ADI $\in [2, 4)$, a perfect demand magnitude mistimed by 2 time-steps yields an average IAE of 0.38, whereas an ADI $\in [16, 30)$ yields a similar score for a mistiming of $\approx 15 - 18$ time-steps. The demand magnitude is assessed consistently across the different subsets. Together, these results show that the metric fulfils its intended purpose: It assesses forecasts for intermittent demands from a perspective of both demand timing and magnitude, while remaining agnostic to demand sizes and inter-demand intervals across time-series, making it usable in a global context.

To show that current metrics cannot assess demand timing and magnitude for heterogeneous datasets, we repeated the experiment using three



(a) ADI: 1.32–2

(b) ADI: 2–4

(c) ADI: 4–7

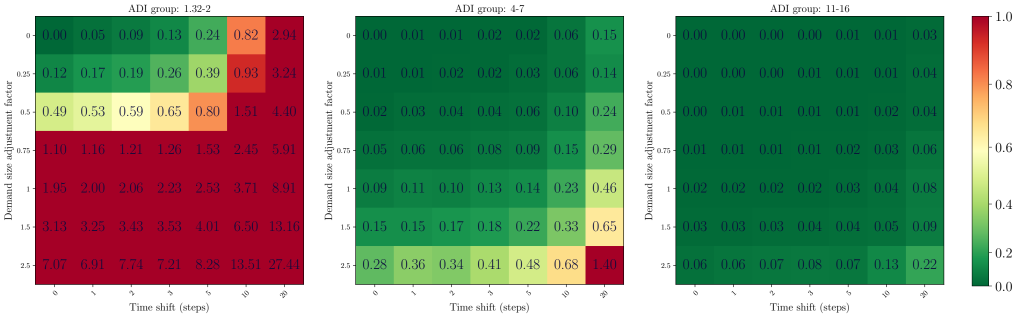
(d) ADI: 7–11

(e) ADI: 11–16

(f) ADI: 16–30

Figure 5: Intermittent alignment error (IAE) across different demand size and time shift modifications for various ADI groups. Each matrix cell shows the average metric score for 500 time-series modified by the demand size (rows) and time shift (columns). The results show that IAE appropriately responds to timing errors based on the level of intermittency, while maintaining a consistent assessment of magnitude errors.

additional metrics: WSMR (rate-based), sMAPE (per-time-step), and SAPE (horizon-wide). The WSMR experiment showed that the metric was consistent for a local context or across homogeneous groups of time-series, yielding progressively worse scores as demand magnitude and timing get worse (Figure 6). However, it fails to show a reasonable progression of errors for mistimed forecasts. Additionally, it is not possible to compare the metric across heterogeneous time-series with differences in ADI; time-series with higher ADI produce lower errors on average. sMAPE is only consistent for perfectly timed forecasts on homogeneous datasets or local models (Figure 7) and should therefore be avoided for intermittent time-series forecasting. SAPE can act as a sanity check for whether models correctly forecast total demand across the horizon, however, it ignores timing entirely (Figure 8).



(a) ADI: 1.32–2

(b) ADI: 4–7

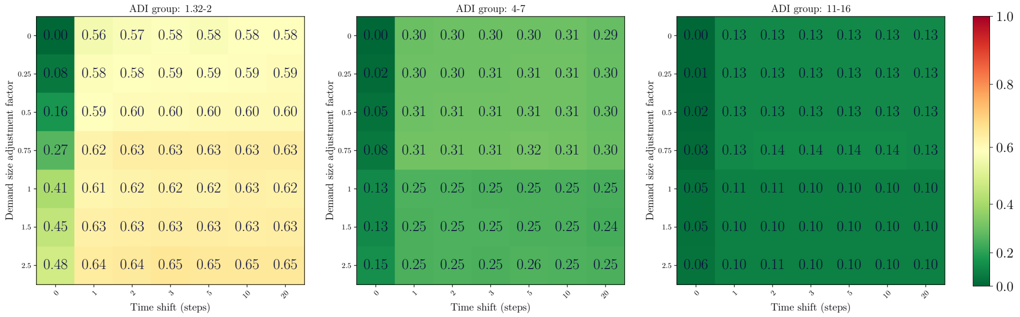
(c) ADI: 11–16

Figure 6: Weighted mean square rate (WMSR) metric performance across demand size and time shift modifications for selected ADI groups. Each matrix cell shows the average metric score for 500 time-series modified by the demand size (rows) and time shift (columns). The metric is consistent for local forecasts or homogeneous groups of time-series but fails to generalise in a global context across heterogeneous datasets with different levels of intermittency.

4.2. Model performance

4.2.1. Main forecasting experiment results and analysis

We compared our approach to two statistical methods, TSB and NPTS, using a dataset with high intermittency ($ADI > 5$) as detailed in Section 3.4.2. Our approach outperformed TSB and NPTS both in terms of demand timing and magnitude (IAE metric), and when treating the entire forecasting horizon as an aggregate forecast (SAPE metric) (Table 2). Interestingly, the SAPE metric shows a worse mean than median score, and

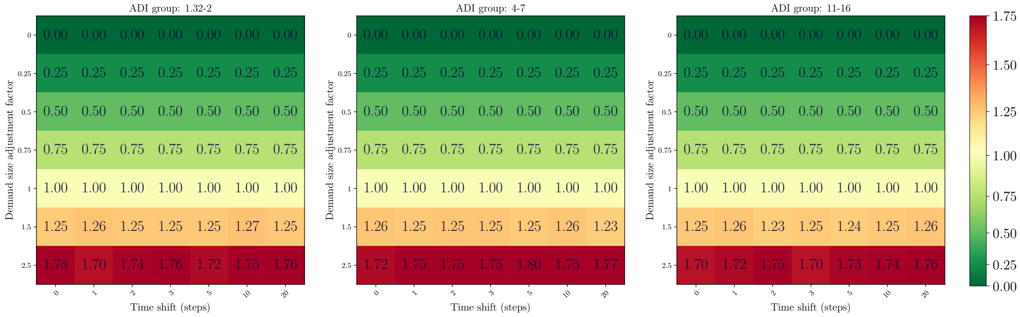


(a) ADI: 1.32–2

(b) ADI: 4–7

(c) ADI: 11–16

Figure 7: Symmetric mean absolute percentage error (sMAPE) performance on synthetic forecasting scenarios with varying demand sizes and time shifts. Each matrix cell shows the average metric score for 500 time-series modified by the demand size (rows) and time shift (columns). Because it only evaluates errors for the same time-step, it is only consistent for perfectly timed forecasts for homogeneous datasets.



(a) ADI: 1.32–2

(b) ADI: 4–7

(c) ADI: 11–16

Figure 8: Scaled absolute percentage error (SAPE) performance under synthetic time shift and demand size adjustments. Each matrix cell shows the average metric score for 500 time-series modified by the demand size (rows) and time shift (columns). The metric only assesses the total forecasted sum across the entire horizon. It can be useful as a sanity check for both local and global models but does not assess the demand timing in any way.

the IAE metric shows a worse median than mean score. This indicates that for most time-series, forecasting the total demand sum is relatively easy, but that difficult edge cases increase the average error. In Figure 9, we show typical forecasts from the three approaches.

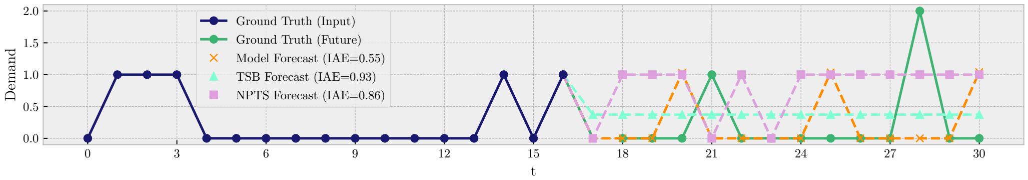


Figure 9: Typical forecasts from our approach, NPTS, and TSB, visualised together for a sample time-series with its ground truth input and future.

Method	**IAE**		**SAPE**	
	Median	**Mean**	**Median**	**Mean**
TSB	0.90	0.90	0.69	1.28
NPTS (75%)	1.00	0.78	1.00	1.58
Our approach	**0.79**	**0.67**	**0.47**	**0.83**

Table 2: Results from the main forecasting experiment comparing our approach to TSB and NPTS (75th percentile). All models used a context length of 5 demands and ADI ≥ 5. Lower values indicate better performance for both IAE and SAPE.

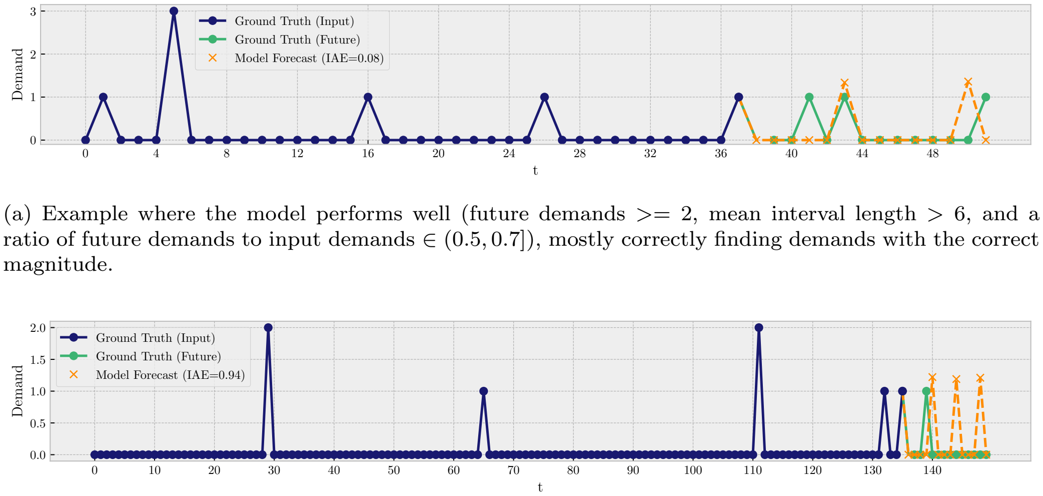
From the exploratory feature importance analysis described in Section 3.4.2, by far the strongest predictor of time-series forecasting performance was the number of future demands (Table 3). Inspection of the learned decision tree structure revealed that time-series with multiple future demands, a high mean interval length, and where future demands were roughly half of the input demands produced the best forecast scores (Figure 10a). In contrast, time-series with only 1 future demand and a long input window were difficult for the model to forecast well (Figure 10b). Additionally, samples with no demands in the forecasting horizon produced large errors because any forecasted demand would yield an IAE of 1.

4.2.2. Varying ADI experiment results and analysis

The performance of our approach was relatively consistent across datasets with different lower ADI thresholds when filtering both training and test

Feature	**Importance (%)**
Number of future demands	75.45
Mean interval length	13.65
Ratio of future to input demands	7.31
Input time-step length	2.17
Future demand sum	0.86

Table 3: Top feature importances from a decision tree model predicting forecasting model performance (IAE) for our approach from time-series characteristics.



(a) Example where the model performs well (future demands >= 2, mean interval length > 6, and a ratio of future demands to input demands $\in (0.5, 0.7]$), mostly correctly finding demands with the correct magnitude.

(b) Example where the model performs poorly (future demands <= 1, input time-steps > 100), over-predicting demands for horizons with few demand occurrences.

Figure 10: Examples of typical scenarios in which our model performs well and poorly.

data (Table 4). TSB and NPTS were more affected by different levels of intermittency than our approach: TSB performed reasonably well with low intermittency with the IAE metric, because timing is less important at lower intermittency, as demands occur more frequently. Additionally, TSB performed best of the three approaches when it came to finding the correct sum across the entire forecasting horizon (SAPE) for the lowest intermittency threshold. As the ADI increased, TSB performed worse. The opposite could be observed for NPTS with the IAE metric: For the highest intermittency it outperformed our approach on average. However, its median performance was very poor, hinting that although some samples were very easy, most were difficult for it. Our approach achieved the most balanced performance across the different time-series characteristics.

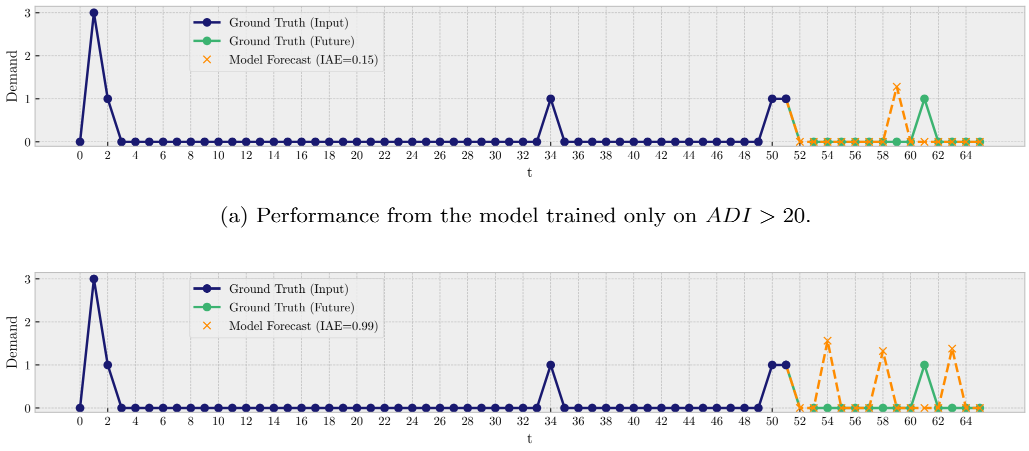
ADI ≥	Train size (our approach)	Our approach				TSB				NPTS (75%)			
		IAE		SAPE		IAE		SAPE		IAE		SAPE	
		Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean
1.32	2 583 968	**0.79**	**0.69**	0.81	1.27	0.85	0.76	**0.46**	**0.91**	0.83	0.73	0.87	1.25
3	552 561	**0.76**	**0.66**	**0.48**	**0.99**	0.88	0.86	0.67	1.34	0.92	0.79	1.00	1.80
7	164 823	**0.81**	**0.68**	**0.46**	**0.67**	0.90	0.91	0.67	1.10	1.00	0.73	1.00	1.32
12	42 079	**0.83**	0.67	**0.44**	**0.50**	0.93	0.93	0.70	0.87	1.00	**0.59**	1.00	1.10
20	6 275	**0.83**	0.60	1.00	**0.67**	0.96	0.94	**0.75**	0.77	1.00	**0.52**	1.00	1.03

Table 4: Performance of each forecasting method on subsets of the dataset with increasing ADI thresholds. A context length of 5 demands was used for all models. Lower values are better for both intermittent alignment error (IAE) and sum aggregate percentage error (SAPE).

To evaluate our approach from a global modelling perspective, we examined whether performance improved when the training data more closely matched the characteristics of the test data. We trained one model on all training samples ($ADI > 1.32$) and another solely on very intermittent samples ($ADI > 20$), then tested both on the high-intermittency test set ($ADI > 20$). The model trained on more similar data achieved better performance in terms of IAE (Table 5). For SAPE, however, the mean performance was slightly worse and the median significantly worse. This is partly because the $ADI > 20$ model more often failed to predict any demand within the forecasting horizon, resulting in a SAPE of 1. In contrast, the $ADI > 1.32$ model tended to overpredict, rarely producing empty forecasts. These findings suggest that training semi-global models on more homogeneous subsets may yield better results than fully global models. Figure 11 illustrates this difference, showing that the global model frequently overpredicted demand, likely due to exposure to less intermittent time-series during training.

Training data (ADI threshold)	Train size	IAE		SAPE	
		Median	Mean	Median	Mean
ADI > 1.32	2 583 968	0.91	0.73	**0.41**	**0.63**
ADI > 20	6 275	**0.83**	**0.60**	1.00	0.67

Table 5: Comparison of two models evaluated on the same test set (ADI > 20) but trained on subsets with different ADI thresholds. A context length of 5 demands was used for both. Lower values are better for both intermittent alignment error (IAE) and sum aggregate percentage error (SAPE).



(a) Performance from the model trained only on $ADI > 20$.

(b) Performance from the model trained on all samples $ADI > 1.32$. The model over-predicts the demand, which is a common occurrence on highly intermittent time-series when using the global model.

Figure 11: Visualisation of differences in model behaviour on the same sample when trained on a homogeneous (semi-global) versus heterogeneous (global) scale.

4.2.3. Varying context length experiment results and analysis

Varying the context length, i.e., the number of historical demands visible to the forecasting model, did not meaningfully affect the performance of our approach. Forecasting performance, as measured by IAE, remained nearly constant across all tested context lengths, and only a slight improvement in SAPE was observed with longer contexts (Table 6). TSB exhibited the greatest sensitivity to context length: its SAPE scores were poor at shorter lengths ($\in 3, 5$) but improved substantially with a longer input. These results suggest that our approach remains robust even when historical demand data are limited, making it a valuable alternative in early product life cycles or in cases where article version history is incomplete.

Context length	Train size (our approach)	Our approach				TSB				NPTS (75%)			
		IAE		SAPE		IAE		SAPE		IAE		SAPE	
		Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean
3	345 518	**0.79**	**0.67**	**0.48**	**0.89**	0.92	0.91	1.30	2.41	1.00	0.80	1.00	1.96
5	331 406	**0.79**	**0.67**	**0.47**	**0.83**	0.90	0.90	0.69	1.28	1.00	0.78	1.00	1.58
10	294 568	**0.79**	**0.67**	**0.46**	**0.78**	0.89	0.90	0.56	0.81	1.00	0.77	1.00	1.26
15	258 724	**0.80**	**0.67**	**0.46**	0.78	0.89	0.90	0.55	**0.77**	1.00	0.76	1.00	1.16
20	227 240	**0.81**	**0.69**	**0.47**	0.81	0.89	0.90	0.56	**0.78**	1.00	0.74	1.00	1.12

Table 6: Performance of each forecasting method at different input context lengths. ADI ≥ 5 was used for all. Lower values are better for both intermittent alignment error (IAE) and sum aggregate percentage error (SAPE).

4.2.4. MTO packaging manufacturing experiment results and analysis

We performed the experiment on the packaging manufacturing data to examine how the modelling approach would perform in a new domain. This dataset differed slightly from the M5 dataset (Table 1), as it was aggregated monthly and had a lower level of intermittency overall. These characteristics made it a suitable candidate for evaluating which modelling strategy to adopt when using the approach in a new domain.

Models trained on the M5 dataset did not generalise as well as those trained entirely on, or fine-tuned with transfer learning to, the target dataset (Table 7). This suggests that greater homogeneity in the training data is beneficial. Interestingly, transfer learning performed slightly worse than training solely on the packaging manufacturing dataset, showing that learning base features from a larger dataset did not give the model an advantage, despite the smaller size of the target training set.

Comparing the results in Table 7 to the main experiment (Table 2), the forecasting performance was better overall on the packaging manufacturing dataset than the M5 dataset, when measured by both IAE and SAPE. One potential explanation for this is the absence of very intermittent time-series with $ADI > 20$. However, the dataset still had considerable variation in its samples but managed to learn both moderate and low intermittency reasonably well in one model (Figure 12).

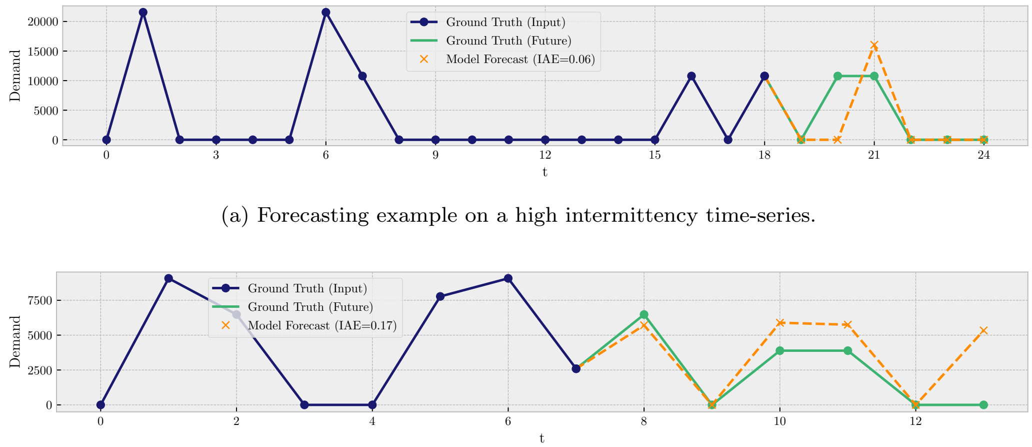
Method	**Training**	**IAE**		**SAPE**	
		Median	**Mean**	**Median**	**Mean**
TSB	–	0.75	0.64	0.43	0.78
NPTS (75%)	–	0.78	0.66	0.83	1.22
Our approach	Packaging only	**0.43**	**0.48**	**0.31**	**0.55**
Our approach	M5 only	0.57	0.54	0.45	0.78
Our approach	M5 + Packaging	0.54	0.53	0.44	0.76
Our approach	Transfer learning	0.45	0.49	0.33	0.60

Table 7: Performance of our approach, TSB, and NPTS (75%) across different training setups for the packaging manufacturing dataset. Lower values are better for both intermittent alignment error (IAE) and sum aggregate percentage error (SAPE).

5. Discussion and future work

The proposed IAE metric is consistent in assessing demand timing and magnitude across heterogeneous datasets, being agnostic to both the average inter-demand interval ranges and scale of demands for different time-series (Figure 5). Unlike SPEC or PIS (Martin et al., 2020; Wallström and Segerstedt, 2010), which are tailored to inventory contexts, or MSR (Kourentzes, 2014), which assesses rate-based local forecasts, the IAE metric provides a consistent point-forecast evaluation across heterogeneous time-series, making it better suited to global forecasting contexts. It can be used as a stand-alone metric both during the development of models when selecting between different hyperparameter configurations, and when choosing between different models.

Some aspects of the metric calculations are ad-hoc, such as the parameters used in the logistic transformation to bound the errors between 0 and 1 (Equation 9), as well as the calculation of the error term (Equation 5). These components were developed to address issues that previously led to unfair



(a) Forecasting example on a high intermittency time-series.

(b) Forecasting example on a low intermittency time-series.

Figure 12: Forecasting examples from the packaging manufacturing model, showing two forecasts on time-series with different characteristics but predicted using the same model, trained solely on packaging manufacturing training samples.

results and were necessary to achieve the fair and robust performance shown in Figure 5. The way the metric balances demand timing and magnitude will always involve a degree of subjectivity; there is no defined importance between these two components, and it differs per case. It would be interesting to extend the metric with separate weight parameters for the timing and magnitude in the future, and let the forecasting task guide their relative importance, perhaps by simulating outcomes and tuning the weights.

TSB produces satisfactory forecasts for time-series with low intermittency ($ADI \lessapprox 3$), but it becomes less useful as intermittency increases due to the absence of a demand timing component. Although TSB and the Croston method have long been used in retail settings where demand flows steadily (Kourentzes, 2013), they fall short in domains such as make-to-order manufacturing, where demand timing is more critical to decision-making. NPTS performed well at identifying timing in highly intermittent cases and occasionally outperformed our model in straightforward scenarios when intermittency was high ($ADI \gtrapprox 12$). However, it lacked robustness and was less accurate in estimating total demand across the forecasting horizon (Table 4). In contrast, our approach demonstrated greater versatility and robustness across different time-series characteristics (Table 4), and it achieved reliable forecasts even with a much shorter context length (Table 6). This capability is especially useful in real-world settings where articles are frequently replaced or newly introduced, enabling forecasting earlier in the product lifecycle. For the packaging manufacturing dataset used in this study, forecasts became reliable after only 3 demands (on average, ≈ 6 months; Table 1), whereas TSB required around 10 demands (≈ 20 months) to reach similar robustness. Although the best-performing context length for our model (as measured by SAPE) was 10 historical demands, performance differences across context lengths were negligible when measured by IAE.

The median IAE score was consistently worse than the mean for our approach (Table 2, Table 4, Table 5, Table 6, Table 7). This indicates that the majority of samples are difficult for the model to forecast well, but that it performs very well for some cases. Therefore, model training should be followed by an analysis procedure to understand which cases the model handles well, and which ones warrant less trust in the forecasts. An interesting direction for future work is to develop models that can output per-sample confidence thresholds. Another approach could involve identifying time-series features which lead to good or bad forecasting performance to provide additional decision support to practitioners using the forecasts. This study serves as a first step in showing how we can decompose, pre-process, and structure intermittent time-series to train global models, but more work is needed to produce good forecasts for all types of intermittent time-series.

Although our approach is robust to different levels of intermittency (Table 4) and different context lengths (Table 6), there is a clear difference in forecasting performance when using a model trained on data more similar to the target task (Table 5, Figure 11). This indicates that the number of samples is less important than training data composition, despite transformers being known to be data hungry (Zeng et al., 2023). Previous studies have found that semi-global models produce better results than global models (Bandara et al., 2020; Oriona et al., 2023; Abbasimehr and Noshad, 2025; Sonnleitner, 2025), which seems to hold true for our proposed approach as well. Because tuning hyperparameters for local models across thousands of time-series is infeasible due to time and complexity, it is worth exploring semi-global models further: What level of homogeneity is required? How do we best separate datasets into training clusters, and based on which factors? When is it worthwhile to use local models or fully global models? Semi-global models would still be able to support the data hungry model types while getting the benefits of homogeneity that local models have.

The forecasting approach transferred well to a different domain than the one it was developed on (Table 7, Figure 12). Training exclusively on the new domain data yielded the best results, despite the dataset being significantly smaller, lending additional support to the use of homogeneous training subsets over fully global datasets. Training solely on the M5 dataset did not yield a sufficiently general model to be used as a foundation model for new domains. The lack of diversity in public datasets is one of the main challenges in constructing global foundation models for time-series forecasting (Goswami et al., 2024; Liang et al., 2024).

Because we decompose intermittent data into separate components for demand magnitude and timing, our forecasting approach assumes that predictions are being made at the time-step in which the last demand occurred. As a result, the model treats the time-step immediately after a demand and one following a prolonged gap of zero demand as equivalent, which is a limitation we aim to address in the future, potentially by supplying the number of zero demand steps as an exogenous variable to the model.

Other exogenous variables could also be included to inform better forecasts but the best way to incorporate them in our model remains an open problem; relevant external factors vary across different parts of the dataset, which could render the global modelling approach less effective.

Furthermore, validating the approach in a real-world setting is an important next step, for instance, by incorporating the forecasts into decision support systems in make-to-order (MTO) manufacturing, where accurate demand timing and magnitude predictions directly influence purchasing and production decisions.

6. Conclusions

This paper has presented a global demand forecasting approach for intermittent time-series, producing point forecasts that capture both demand timing and magnitude at each time-step. A series of pre-processing steps were developed to reduce skewness in the data distribution and make it more suitable for processing by deep learning models. The proposed forecasting approach outperformed both demand rate and non-parametric methods in most scenarios, both when taking timing into account, as well as when observing the entire forecasting horizon. The experiment results results showed that the global deep learning approach can generalise across heterogeneous collections of intermittent time-series. Both demand timing and magnitude could be forecasted to an extent. However, many open challenges remain for improving the performance further.

We also proposed a new metric, Intermittent Alignment Error (IAE), to assess timing-aware intermittent forecasts across collections of heterogeneous time-series, agnostic to individual time-series' level of intermittency and demand magnitudes. We performed an experiment which demonstrated that IAE is robust across different scenarios and consistently evaluates both demand timing and magnitude, adjusting its tolerance for timing errors based on the level of intermittency.

The forecasting approach is a first attempt at generating structured point forecasts that capture both demand timing and magnitude across the entire forecasting horizon. Several open challenges remain to improve the forecasts, e.g., determining the appropriate level of heterogeneity for semi-global models, how to best include exogenous parameters, and how to quantify forecasting uncertainty. Addressing these open challenges will be important to enhance the practical relevance of our approach for decision-making. We hope that this study provides a foundation for future work on forecasting intermittent demand with both timing and magnitude, and that the proposed metric serves as a useful benchmark for evaluating such forecasts in a global context with heterogeneous data.

Acknowledgements

Funding: This work was supported by the KKS foundation through the S.E.R.T. Research Profile project www.rethought.se at Blekinge Institute of Technology).

Thank you to Oskar Pettersson and Conrad Tingström for valuable discussions during the technical development of this study.

Appendix A. Hyperparameter tuning

Hyperparameter	**Range (Run 1)**	**Range (Run 2)**	**Best Value**	**Description**
Dataset-related				
Batch size	64–2048	1024–2048	1628	Number of samples per training batch.
Bin balance factor	{1, 2}	{1.5, 2, 2.5, 3}	1.5	Max size ratio between largest and smallest bin during undersampling.
Number of bins	10–30	5–15	7	Number of bins used for target undersampling.
Gaussian noise std	0.00–0.60	0.10–0.75	0.36	Standard deviation of Gaussian noise added to inputs during training.
Keep outliers	{True, False}	{True, False}	True	Whether to reintroduce extreme values after undersampling.
Lower percentile	0.01–0.20	0.06–0.20	0.08	Lower bound percentile for defining outliers.
Upper percentile	0.80–0.99	0.80–0.94	0.92	Upper bound percentile for defining outliers.
Model architecture				
d_{model}	64–1024	16–256	144	Transformer dimension (must be divisible by head count).
Number of heads	2–8	2–8	2	Number of attention heads in the transformer encoder.
Encoder layers	2–6	2–6	2	Number of transformer encoder layers.
Feedforward dim	32–512	256–1024	831	Size of the hidden feedforward layer.
Dropout	0.0–0.6	0.0–0.5	0.24	Dropout applied after attention and feedforward layers.
Input layer norm	–	{True, False}	False	Whether to apply layer normalisation before the transformer input.
Training and optimisation				
Learning rate	10^{-6}–10^{-3}	10^{-6}–10^{-4}	$1.60 \cdot 10^{-6}$	Initial learning rate for the optimiser.
Weight decay	10^{-6}–10^{-2}	10^{-6}–10^{-3}	$1.76 \cdot 10^{-6}$	L2 regularisation strength.
Loss function	{L1, MSE}	{L1, MSE}	L1	Loss function used during training.
Scheduler factor	0.1–0.9	0.3–0.7	0.62	Factor by which learning rate is reduced on plateau.
Scheduler patience	5–25	15–30	26	Number of epochs to wait before reducing learning rate.

Table A.8: Hyperparameter random search space across two tuning rounds and the selected best configuration.

Appendix B. Loss curve

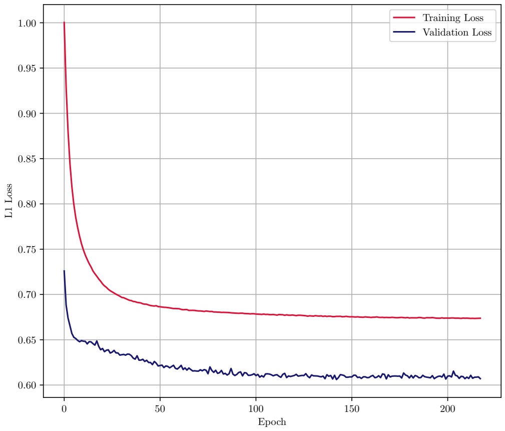


Figure B.13: Training and validation loss curves for the main forecasting experiment.

References

Abbasimehr, H., Noshad, A., 2025. Localized Global Time Series Forecasting Models Using Evolutionary Neighbor-Aided Deep Clustering Method. Journal of Forecasting , for.3263doi:`10.1002/for.3263`.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D.C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A.C., Wang, Y., 2020. Gluonts: Probabilistic and neural time series modeling in python. Journal of Machine Learning Research 21.

Bandara, K., Bergmeir, C., Smyl, S., 2020. Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. Expert Systems with Applications 140, 112896. doi:`10.1016/j.eswa.2019.112896`.

Box, G.E.P., Jenkins, G.M., 1970. Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco.

Croston, J., 1972. Forecasting and stock control for intermittent demands. Journal of the Operational Research Society 23, 289 – 303. doi:`10.1057/jors.1972.50`.

Goswami, M., Szafer, K., Choudhry, A., Cai, Y., Li, S., Dubrawski, A., 2024. Moment: A family of open time-series foundation models, in: Proceedings of the 41st International Conference on Machine Learning, p. 16115 – 16152.

Han, Q., Meng, F., Hu, T., Chu, F., 2017. Non-parametric hybrid models for wind speed forecasting. Energy Conversion and Management 148, 554–568. doi:10.1016/j.enconman.2017.06.021.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735 – 1780. doi:10.1162/neco.1997.9.8.1735.

Jeon, Y., Seong, S., 2022. Robust recurrent network model for intermittent time-series forecasting. International Journal of Forecasting 38, 1415–1425. doi:10.1016/j.ijforecast.2021.07.004.

Kiefer, D., Grimm, F., Bauer, M., van Dinther, C., 2021. Demand forecasting intermittent and lumpy time series: Comparing statistical, machine learning and deep learning methods, in: Proceedings of the 54th Hawaii International Conference on System Sciences (HICSS 2021). doi:10.24251/HICSS.2021.172.

Kourentzes, N., 2013. Intermittent demand forecasts with neural networks. International Journal of Production Economics 143, 198–206. doi:10.1016/j.ijpe.2013.01.009.

Kourentzes, N., 2014. On intermittent demand model optimisation and selection. International Journal of Production Economics 156, 180–190. doi:10.1016/j.ijpe.2014.06.007.

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., Pan, S., Wen, Q., 2024. Foundation models for time series analysis: A tutorial and survey, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, p. 6555 – 6565. doi:10.1145/3637528.3671451.

Makridakis, S., Spiliotis, E., Assimakopoulos, V., 2022. M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting 38, 1346 – 1364. doi:10.1016/j.ijforecast.2021.11.013.

Martin, D., Spitzer, P., Kühl, N., 2020. A new metric for lumpy and intermittent demand forecasts: Stock-keeping-oriented prediction error costs, in: Proceedings of the 53rd Annual Hawaii International Conference on System Sciences (HICSS-53). doi:10.24251/HICSS.2020.121.

Oriona, A.L., Manso, P.M., Fernández, J.A.V., 2023. Time series clustering based on prediction accuracy of global forecasting models. doi:10.48550/arXiv.2305.00473.

Pereira, V., 2021. pyinterdemand - intermittent demand library. URL: https://github.com/Valdecy/pyInterDemand. accessed: 2025-04-25.

Rožanec, J.M., Fortuna, B., Mladenić, D., 2022. Reframing Demand Forecasting: A Two-Fold Approach for Lumpy and Intermittent Demand. Sustainability 14, 9295. doi:10.3390/su14159295.

Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting 36, 1181–1191. doi:10.1016/j.ijforecast.2019.07.001.

Shale, E., Boylan, J., Johnston, F., 2006. Forecasting for intermittent demand: The estimation of an unbiased average. Journal of the Operational Research Society 57, 588 – 592. doi:10.1057/palgrave.jors.2602031.

Sonnleitner, B., 2025. Measuring time series heterogeneity for global learning. Expert Systems with Applications 270, 125666. doi:10.1016/j.eswa.2024.125666.

Syntetos, A.A., Boylan, J.E., Croston, J.D., 2005. On the categorization of demand patterns. Journal of the Operational Research Society 56, 495–503. doi:10.1057/palgrave.jors.2601841.

Teunter, R.H., Syntetos, A.A., Zied Babai, M., 2011. Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research 214, 606–615. doi:https://doi.org/10.1016/j.ejor.2011.05.018.

Türkmen, A.C., Januschowski, T., Wang, Y., Cemgil, A.T., 2021. Forecasting intermittent and sparse time series: A unified probabilistic framework via deep renewal processes. PLOS ONE 16, e0259764. doi:10.1371/journal.pone.0259764.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems, p. 5999 – 6009.

Wallström, P., Segerstedt, A., 2010. Evaluation of forecasting error measurements and techniques for intermittent demand. International Journal of Production Economics 128, 625–636. doi:10.1016/j.ijpe.2010.07.013.

Yang, Y., Zha, K., Chen, Y.C., Wang, H., Katabi, D., 2021. Delving into deep imbalanced regression, in: Proceedings of Machine Learning Research, p. 11842 – 11851.

Zeng, A., Chen, M., Zhang, L., Xu, Q., 2023. Are transformers effective for time series forecasting?, in: Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI 2023, p. 11121 – 11128. doi:10.1609/aaai.v37i9.26317.