Spare Parts Prediction: A Comparative Analysis of Croston and its variations, Willemain, Machine Learning, Hybrid, and Combined Methods

Tianhao Chen (477518)

Supervisor: Prof. dr. ir. R. Dekker

Second Assesor: Dr. R. Karpienko

Master Thesis

Data Science and Marketing Analytics

Erasmus School of Economics

Erasmus University Rotterdam

31/07/2023

Abstract

Spare parts have erratic, lumpy, and intermittent demand patterns, characterized by infrequent demand occurrences and multiple extended periods of zero demand. Accurate forecasting and planning are crucial to finding the optimal balance between stock-out and holding costs. This research paper evaluates various forecasting methods applied to four simulated data sets, each representing a unique demand pattern (erratic, lumpy, smooth, and intermittent), as well as four real industrial data sets. The evaluated methods include Croston's method, the Syntetos-Boylan approximation (SBA), the Teunter-Syntetos-Babai method (TSB), Willemain's bootstrapping approach, the hybrid model of Exponential Smoothing and Recurrent Neural Network (ESRNN), two simple-average methods (ensemble 1 and 2), two stacking combinations (Meta-learners 1 and 2), and two machine learning methods (LSTM and RNN). Moreover, SBA is applied to aggregated simulated and industrial data. The results show that the stacking model demonstrates the best overall accuracy performance across all eight data sets, while Willemain shows the most favorable inventory control assessment in simulated data sets and one of the four industrial data sets. Croston and its variation and two stacking methods show the best inventory performance in the other three industrial data sets.

Contents

1 Introduction

Spare parts are components of a machine or system that are specifically designed to replace failed or worn-out parts and are important for maintaining the functionality. When part malfunctions or becomes inoperative, spare parts are utilized to restore the equipment to its operational state, minimizing downtime and ensuring continuous operation. Therefore, spare parts are crucial for maintaining and repairing machines or systems, ensuring operational continuity, and minimizing downtime (van der Laan et al., 2014; Durlinger & van Houtum, 2017).

Poor inventory management resulting in stock-outs of spare parts, which can be costly for industries such as aviation, automotive, and heavy machinery. These industries rely heavily on the availability of their capital goods to produce their manufactured products. The negative consequences of machinery failure or long periods of downtime can result in lower product turnover, leading to substantial financial losses. To ensure efficient handling of machinery shutdowns, companies typically maintain all essential spare parts in stock. However, stocking spare parts during periods of low demand can be expensive, given that spare parts demand is characterized as lumpy, erratic, and intermittent. This implies infrequent demand with many periods of zero demand. Holding spare parts carries the risk of devaluation, price protection costs, and the possibility of the parts becoming obsolete. Companies must find an optimal balance between the inventory holding costs and the potential revenue loss due to equipment downtime. Industrial companies and other businesses must prioritize effective inventory management of their spare parts to avoid stock-outs and subsequent financial losses. The optimal balance between inventory holding costs and potential revenue loss can be achieved through careful planning and forecasting of spare parts demand. This requires an understanding of the various factors affecting spare parts demand, including lumpy, erratic, and intermittent demand, which highlights the need for a data-driven approach to inventory management. Therefore, accurate forecasting of spare parts demand is essential due to its uncertain and time-varying nature, influenced by fluctuations in machine usage and quantities.

Traditional time series forecasting methods, such as the simple moving average (SMA) and simple exponential smoothing (SES), demonstrate poor performance when applied to intermittent demand, as highlighted by Syntetos and Boylan (2005). This inadequacy arises from methodological issues associated with zero values, particularly when dealing with spare parts patterns that exhibit a non-continuous series and these methods tend to rely heavily on the most recent observations.

In the past, only a few time series methods have been developed to forecast intermittent demand. Croston (1972) and its variants, such as SBA (Syntetos, Boylan, 2005) and TSB (Teunter et al., 2011), outperform standard exponential smoothing as it is not well-suited for prolonged periods of zero demand. Willemain et al. (2004) introduced a modified bootstrap approach for inventory management, which has been claimed as having superior performance compared to Croston and its variations. Recently, the M4 and M5 time series competitions have demonstrated the effectiveness of combining statistical models, pure machine learning methods in predicting time series, and hierarchical data aggregation techniques to improve accuracy. The term 'ensemble' is often used to describe the practice of combining two or more forecasting methods with the goal of improving overall prediction performance. By integrating multiple outputs generated by each individual method, ensemble approaches aim to achieve superior forecasting outcomes compared to using a single method alone. The applied methods in M4 and M5, such as hybrid modeling, ensemble modeling through equal weight and optimizer, recurrent neural network methods, and data aggregation, have not been extensively tested on intermittent demand data or evaluated on the characteristics of different types of data sets. Nikolopoulos et al. (2014) conducted a temporal aggregation technique on aviation data by aggregating observations backward at different levels. However, this technique has not been further tested on other industrial or simulated data sets. Moreover, in the M5 competition, participants were not allowed to use Croston and its variations or temporal aggregation method as it was already included in the benchmark. Considering the intermittent nature of the data that the methods are applied to, incorporating models that leverage the Croston's method or temporal aggregation method can potentially lead to a higher accuracy performance.

This investigation leads to the formulation of the following research questions:

1. Do ensemble models outperform Croston and its variations in terms of forecasting accuracy and inventory performance on 4 simulated and 4 industrial spare parts demand data?

To address the main research question, our objective is to improve the accuracy of forecasting intermittent demand by utilizing methods employed in the M4 and M5 forecasting competitions, while using Croston and its variations as a benchmark. We will apply these methods to four simulated data sets, each characterized by a unique pattern such as erratic, lumpy, smooth, and intermittent. This is done to determine whether these characteristics influence the performance of the methods. Furthermore, the same methods will be applied to four industrial data sets to compare the results obtained from the simulated data. Specifically, the focus will be on forecasting spare parts demands and evaluating the performance of both forecasting accuracy and inventory performance.

The sub-research question is:

2. Does aggregating observations into larger intervals improve the accuracy of the forecasting model?

To address the second research question, the training data from simulated and industrial data sets will be aggregated at lower frequency levels. This aggregated data will be used to train the forecasting models. The predictions generated by the models will then be disaggregated back to the original seasonality level for evaluation and compared to the original data without any aggregation.

This paper will use Croston and its variations as the benchmark: Croston, Croston with an optimized smoothing factor, Teunter-Syntetos-Babai (TSB), and the Syntetos-Boylan Approximation (SBA). The methods to be explored are the high-performing models and derived combining techniques in the M4 and the M5 Forecasting Competition: ESRNN, combined Croston-based models through equally distributed weight and via stacking, recurrent neural network (RNN), and Long Short Term Memory (LSTM). These models will be trained on various types of data sets to determine their accuracy and inventory control performance. To further validate and compare the results, Willemain's bootstrapping approach will also be incorporated. In the final step, the results will be discussed with the existing literature to discuss their differences.

This research is academically significant as it aims to expand the knowledge of spare parts forecasting by exploring the combination method, hybrid method, data aggregation, and recurrent neural network. The results obtained will allow for comparisons of accuracy improvements compared to Croston, SBA, and TSB. Additionally, the practical relevance lies in assessing whether these models provide better inventory performance. The paper will start with a brief review of the existing literature on spare parts forecasting, followed by a detailed explanation of the methodology, introduction of simulated and industrial data sets, the chosen accuracy measures, and inventory control considerations. Upon comparing the results, conclusions are drawn to address the research questions, along with a discussion of limitations.

2 Literature About Spare Parts and Forcasting

Spare parts are stock items that keep machinery activities up or keep products processed in optimal conditions (Kennedy et al., 2002). However, when it comes to forecasting, it is twice as difficult as traditional time series forecasting as the spare parts demand is lumpy, erratic, and intermittent in nature (Petropoulos et al., 2014). High sources of uncertainty in intermittent forecasting need to be dealt with regarding the volatility in demand sizes, timing of demand, and long periods without any demand. To make things worse, time series components as trend, level, and seasonality should not be considered to be evident as these features are difficult to detect in spare parts demand patterns (Petropoulos et al., 2014).

Furthermore, according to Syntetos et al. (2016), nearly 60% of any existing inventory consists of stocks shared in intermittent characteristics, which becomes obsolete when time decays as no demand occurs. Therefore, it is a challenging task for companies who are dealing with their spare parts management to find the optimal balance between the probability of stock out and the inventory holding costs.

This section begins with a comprehensive overview of spare part forecasting methods and the insights gained from the M4 and M5 forecasting competitions. Following that, we present a thorough review of prior comparison studies that haven compared various forecasting methods for spare parts demand, taking into account among others methodologies, data sets, and evaluation criteria. Finally, based on the insights obtained from the M4 and M5 competitions and the findings of previous comparison studies, a conclusion is derived regarding the forecasting methods to be implemented in this research.

2.1 Intermittent Forecasting Methods

The conventional forecasting methods such as SMA and SES fail to perform well for intermittent demand (Syntetos and Boylan, 2005) due to methodological problems from the zero values as spare parts pattern is considered as non-continuous series. The first defining approach was introduced by Croston (1972), in which the non-zero demand interval and its demand size were separately estimated by exponential smoothing. According to Willemain et al. (1994), Croston with positive demand was robustly superior to SES in terms of smoothness, variation, and average inventory holding costs while in the same service level. Since then, Croston has been used as a benchmark as it generates lower safety stock at the same service level. Consequently, Syntetos and Boylan (2005) modified Croston by introducing a bias correction. According to several studies (Syntetos and Boylan, 2006; Teunter et al., 2011; Zhu et al., 2017; Babai et al., 2019), Croston outperformed SBA in terms of service levels whereas SBA was more accurate on different types of data sets. However, when it comes to obsolescence, i.e., demand decreases over time toward zero, both models perform poorly as they don't measure the obsolescence when the gradual decreasing demand occurs. Teunter et al. (2011) approached the issue by forecasting the demand probability while updating the demand when zero demand occurred. More specifically, the demand forecast is adjusted downward when there is no demand to detect obsolescence. However, when the model was tested empirically (Babai et al., 2019), there were no significant improvements.

2.2 Paramatric and Non-Parametric Intermittent Forecasting Methods

Croston and its variations are parametric approaches that assume the data will follow a predefined probability distribution. Croston (1972) assumes that the demand interval follows the geometric distribution while the demand size per demand interval follows a normal distribution. On the other hand, Bootstrapping techniques such as Willemain, Empirical Method (Porras and Dekker, 2008), variants on the empirical method, and neural network models are categorized as non-parametric approaches, whereas no probability distribution function is assumed. Parametric approaches may previously be inferior to non-parametric methods in determining the best lead demand distribution since the distribution is not always distributed parametrically, and by selecting an incorrect distribution it may lead to poor results. Because of this, non-parametric approaches have been devised, and apply techniques looking for empirical ways to explain the lead-time demand. This suggests that non-parametric approaches may be used for a wider range of data than parametric methods, such as data with extremely irregular demand (Smith and Babai, 2011). Several studies (Bookbinder and Lordahl, 1989; Hasni et al., 2019) demonstrate that utilizing a non-parametric technique limits the possibility of assuming an incorrect distribution, which results in superior estimates.

For example, Willemain et al. (2004) estimated intermittent inventory data with autocorrelation by simulating zero and non-zero requests using a two-state Markov process to capture the autocorrelation, and then assessed the probabilities directly from the data by utilizing the counting methods. A sequence of zeros and ones was generated using the transition probabilities. The next step was to produce demand sizes using a jittering process, which modified the sampled demand values by adding random fluctuation. The method was applied to several industrial data sets and the performance was more accurate than Croston or SBA. Another type of bootstrapping is proposed by Zhou and Viswanathan (2011), which produced demand sizes by sampling and demand intervals using bootstrapping. They used simulated and empirical data to compare this strategy to SBA in terms of inventory performance. They found that the SBA was performing poorly when applied to empirical data than to simulated data. In a comparison analysis, Hasni et al. (2019) found that for highly intermittent demand and short lead-times, SBA outperformed the bootstrapping technique in terms of inventory performance. On the other hand, when dealing with moderately intermittent demand and longer lead-times, the proposed method performed better than SBA and Willemain.

In addition to the mentioned methods above, the mainstream methods include also machine learning methods in predicting time series. Machine learning is known for its cross-learning between the predictors to forecast the variable of interest. These techniques seek to identify the underlying dependencies present in the demand data and use these to forecast future values and are capable of capturing complex patterns in long-term forecasts (Hyndman & Khandakar, 2008). One of the first who used machine learning techniques in predicting intermittent demand was Gutierrez et al. (2008), who compared the outcome with SES, Croston, and SBA using electronic distributor data. The results were found to be more accurate. Later on, Kourentzes (2018) extended the modified model proposed by Mukhopadhyay et al. (2012), demonstrating that the neural network technique outperformed Croston and its variations in terms of service levels but performed poorly in accuracy metrics due to the difference between levels in the intermittency and the choice of error function.

2.3 M4 and M5 Forecasting Competitions

M4 and M5 competitions have further expanded the expertise within the time series by the competitors competing against each other to improve the existing models. In M4, the data come with different frequency and seasonality. It is clear that from M4 (Makridakis, Spiliotis, & Assimakopoulos, 2020; Hyndman, 2020), pure combined statistical, machine learning, and/or mixed methods outperform the base statistical or machine learning algorithm. The competition showed that as well as the results of the past three M Competitions were confirmed by the better numerical accuracy of combining statistical and machine learning approaches. It indicates that no single approach is capable of accurately capturing time series patterns, but that a mixture of methods, each of which captures a different aspect of such complex patterns, is more accurate since it leverages the errors of individual models. Syml (2018) won the M4 competition by mixing models of both statistical and machine learning features showing a winning significant 10% improvement over the benchmark. The model can be described as hierarchical nature of parameter selection both globally and locally for each time series, based on the variation of the RNN. Moreover, Montero-Manso et al. (2020) utilized XGBoost optimizing outputs produced by statistical time-series models, and the model was ranked as second behind the hybrid approach.

In addition to M4, the data in M5 consists contextual information, and the sequential data was grouped in a cross-sectional manner, which also shows lumpy, erratic, and intermittent characteristics. Participants may choose their approach to forecast unit sales at different aggregation levels in several sub-groups. The results showed, confirmed the finding of the previous competition, the improved accuracy of combining forecasts from different methods (Makridakis, Spiliotis, & Assimakopoulos, 2022). For example, the winning method used an equal-weighted combination of six models, each of which took use of a distinct training set and learning strategy. Similarly, an equal-weighted combination of five models, each of which had a different estimate of the trend, was employed by the technique that placed second.

Furthermore, the results showed that the top-performing submissions were not superior in all aggregation levels. For example, the winning team was only superior in lower aggregation levels, i.e., low frequency yearly compared to daily data. The same held for the second-ranked team, who was also superior in lower aggregation levels. This can be explained as characteristics like trend and seasonality are challenging to identify when forecasting disaggregated data with unpredictable sales (Kourentzes, Petropoulos et al., 2014). Similarly, Nikolopoulos et al.(2014) applied Aggregate-Disaggregate Intermittent Demand Approach (ADIDA) to British Royal Airforce data and claimed improved accuracy when the data was temporal arrogated into lower aggregation levels, but the error increased when the temporal arrogation was too low.

Lastly, the M5 Forecasting Competition marks a significant shift as participants began heavily incorporating machine learning combination methods, specifically utilizing LightGBM and RNN. These approaches demonstrate their capability in handling a wide range of correlated time series data. All of the top-performing techniques in the competition surpass the performance of traditional statistical benchmarks and their combinations. However, the question raises whether these machine learning methods would also achieve a high accuracy performance in the context of spare parts demand, where the spare parts are largely uncorrelated with each others.

2.4 Comparative Studies

Pinçe et al. (2021) did a quantitative literature analysis on 53 publications providing technique comparisons to summarize the research on estimating intermittent demand. They concluded that the SBA had a higher forecasting performance than Croston in 40 or more research papers while being less clear when compared to inventory control. The comparative studies have been extended by De Haan (2021) and Nguyen (2023) as they benchmarked Croston, TSB, SBA, Willemain, MLP, and LightGBM on four simulated and four industry data sets. They found that from an overall perspective, SBA had the highest accuracy, while Willemain had the highest inventory control. But when there was extreme intermittency, MLP and LightGBM approaches delivered the best inventory control results. For example, when compared to mean absolute scaled error (MASE) using artificially erratic data, SBA performed better than all other methods. However, when compared to smoother data, Croston was superior in terms of mean squared error (MSE) and rooted mean squared scaled error (RMSSE). In contrast, statistical methods were more superior when used to forecast industrial data while MLP was superior in forecasting simulated smooth data. Also, it is good to know that the LightGBM performed the worst among all the models in terms of accuracy.

2.5 Conclusion Literature Review

Both M4 and M5 Forecasting Competitions have highlighted the potential of combining statistical and machine learning methods to improve forecasting performance beyond that of individual models. However, these findings are not extensively explored or tested on different types of demand. The research conducted by De Haan (2021) revealed that the accuracy performance of Croston and its modified models was influenced by the type of demand, whether it was simulated generated or derived from the industrial data. This raises the question of whether combing Croston and its modified models can exhibit superior accuracy performance across erratic, lumpy, smooth, and intermittent demand patterns, as well as for both generated and industrial data. In this paper, we will construct combined Croston and its variations model. These models will be combined using simple-average (equal-weighted) and a stacking method facilitated by an optimizer. Two pools of methods will be utilized, with the second pool incorporating the forecast of RNN. Therefore, a total of four combined models will be evaluated. Additionally, we will employ the hybrid method ESRNN to assess its contribution to intermittent demand forecasting, as this model has not been previously applied in this context.

Furthermore, the M5 competition demonstrated the significant contribution of the machine learning method RNN and LightGBM. However, in this research, we will exclude LightGBM as it has been previously tested (De Haan, 2021) and found to be only effective in inventory performance under extreme intermittency conditions. Instead, we will assess the performance of two types of RNN: the classic RNN and the LSTM. Finally, it was observed in De Haan (2021) and in Nguyen (2023) that Willemain exhibited the overall best inventory performance. Therefore, we will compare the performance of the combined models with Willemain as well.

3 Methodology

This paper aims to extend the existing comparative studies on spare parts forecasting by evaluating different methods and their performance in forecasting various characteristics of demand. The workflow follows a similar framework conducted by Pinçe et al. (2021), Syntetos (2005), and Teunter et al. (2011). The benchmark includes comparing the proposed approach to Croston, TSB, SBA, and Willemain in terms of accuracy and inventory control metrics such as achieved fill rate and holding costs. In this context, Croston-based model is used to refer the three forecasting methods Croston, TSB, and SBA as these methods are all based on the Croston. In addition to Croston-based models, this study introduces the use of combined Croston-based models, Willemain, two types of recurrent neural networks, hybrid exponential smoothing-recurrent neural networks (ESRNN), and ADIDA .

Furthermore, the setting for each method will be explained regarding the data processing and the forecasting generation. It also covers the hyper tuning of any specific parameters to each method. For all the data sets considered, the first 70% of observations will be utilized as training data, while the remaining 30% will be used for testing. In the case of the stacking method, predictions on the validation data will also be generated, a subset of 10 from the most recent observations from the training data will be reserved for validation purposes in the simulated and OIL data sets, 4 observations for AUTO, 30 observations for MAN, and 15 observations for BRAF.

3.1 The Croston's method

The first who recognized this is Croston (1972), who devised a new approach that is now known as Croston. The intermittent demand issues that SES faces have been resolved by Croston. His approach separates the inter-demand period and the demand size into two sections of the demand estimation. It uses SES to forecast these values separately to provide smoothed estimates over time.

Let q_i be the i th non-zero quantity, i.e. demand, and a_i be the estimate of time interval between non-zero demands q_{i-1} and q_i. If $\hat{q}_{i+1|i}$ and $\hat{a}_{i+1|i}$ denotes the one step forecast and using α as smoothing factor, the method gives:

$$\begin{aligned}\hat{q}_{i+1|i} &= (1-\alpha)\hat{q}_{i|i-1} + \alpha q_i, \\ \hat{a}_{i+1|i} &= (1-\alpha)\hat{a}_{i|i-1} + \alpha a_i.\end{aligned} \tag{1}$$

The smoothing parameter α is assumed to have values between 0 and 1, and is equal for both Equations (1). Let j represent the moment of the most recent positive observation. The ratio is then used to calculate the h-step forward prediction for the demand at time $T + h$,

$$\hat{y}_{T+h|T} = q_{j+1|j}/a_{j+1|j}. \tag{2}$$

3.2 The SBA Method

The SBA was proposed by Syntetos and Boylan in 2005 and is similar to Croston in that it separates the demand. A bias correction coefficient is added to Croston. The inter-demand interval is smoothed by parameter a, which aims to lessen the bias:

$$\hat{y}_{T+h|T} = \left(1 - \frac{a}{2}\right) q_{j+1|j}/a_{j+1|j}. \tag{3}$$

SBA generally exceeds Croston in terms of predicting accuracy, whereas Croston performs better in terms of inventory performance given different types of demand patterns (Pinçe et al., 2021). However, SES, Croston, and SBA might perform poorly in the presence of obsolescence or combines with declining or increasing trend.

3.3 The TSB Method

Teunter et al.(2011) created TSB for these unique demand patterns with the presence of obsolescence. The demand size approach is still the same as in Croston, but instead of using the inter-demand interval projection, TSB mixes it with the demand probability estimate. In the absence of demand, this approach adjusts the demand size, whereas Croston's method does not update. This results in a projection being revised downward when there is low demand, which shortens the amount of time required to detect obsolescence. The demand probability d_i is added through SES to Croston in place of the inter-demand interval. When demand is met, it is 1, otherwise it is 0:

$$\hat{y}_{T+h|T} = p_{j+1|j} q_{j+1|j}. \tag{4}$$

3.4 Setting Croston, TSB, and SBA

The Croston, TSB, and SBA models in this study are implemented using the Statforecast library (Statforecast, 2023). The library includes also an optimized Croston model that tunes the smoothing parameters based on the average demand size and interval within the range of 0.1 to 0.3 to optimize the performance. Additionally, for the TSB model, the smoothing parameters for the average demand size and its probability must be specified. Therefore, the parameters are manually optimized within the range of 0.1 to 0.9, with steps of 0.2.

3.5 Willemain

Willemain et al. (2004) developed a modified bootstrap approach, that can deal with frequently repeated value, autocorrelation, and short series. The model starts with a two-state, Markov process to produce a series of zero and non-zero values over the L period of the lead-time. More specifically, the forecast depends on the last observation whether the value is zero or non-zero. The next step is to change the non-zero forecast by giving a specific numerical value. Instead of resampling from historical non-zero-value, which would be both unrealistic and overfitting. They chose to add some bias to capture more variance from a lumpy, erratic, and intermittent demand:

$$JTTERED = 1 + \mathrm{INT}\left\{X^* + Z\sqrt{X^*}\right\}. \tag{5}$$

Where X^* denotes any random historical observation, and Z denotes bias, only the non-zero estimates will be jittered. One prediction of the lead-time demand (LTD) is produced once the expected values are added up over the forecast horizon. Lastly, estimating transition probabilities and jittering are repeated to get multiple predictions, and these values are then sorted to generate a distribution of the LTD.

3.5.1 Setting Willemain

To replicate the results outlined by Willeman et al. (2004), the process described by De Haan (2021) is conducted. Firstly, we generate the transition probabilities for the two-state Markov chain. The Markov chain generated 0s and 1s over the specified forecast horizon. The non-zero demand values are then modified with random values from the historical data. Next, a jittering technique is applied to the non-zero demand values. After the fitting process, the final forecast is calculated as the mean of the distribution of lead-time demand.

3.6 The Recurrent Neural Network Method

RNN uses the idea of saving memory, which enables them to save the states or details of prior inputs in order to produce the subsequent output in the sequence. Recurrent neural networks' outputs are reliant on the previous parts in the sequence, unlike typical deep neural networks, which presumes that inputs and outputs are independent of one another. The network computes the values of the hidden units and the final output after k time steps in the feedforward layer. The networks' related weights are shared throughout time (Elman, 1990). There are two types of weights for each recurrent layer: one for the input and the other for the hidden unit. Similar to an ordinary layer in classic neural networks, the final feedforward layer computes the final output for the k-th time step. The structure of RNN can be explained as follow:

Let's say we have k hidden layers. At time step t, the input $x_t \in R$ is assumed to be a one scalar value feature, $w_x \in R^{inputs}$ are weights associated with inputs in the recurrent layer and $w_h \in R^{hidden}$ are weights associated with hidden units in the recurrent layer. The current context $h_t \in R^m$ vector stores the values of the hidden units/states at time t, m is the number of hidden units. Initial stage the feedback value h_0 is set to zero. $b_h \in R^m$ serves the associated bias associated within k-th layer, and $y_t \in R$ is the output of the network at time step t.

At every k-th time step, we compute the h_t with the use of an activation function until the unfolding process, i.e. the feedforward layer:

$$h_{t+1} = f\left(x_t, h_t, w_x, w_h, b_h\right) = f\left(w_x x_t + w_h h_t + b_h\right). \tag{6}$$

The output y at time t is computed as:

$$y_t = f\left(h_t, w_y\right) = f\left(w_y \cdot h_t + b_y\right). \tag{7}$$

In other words, during the training phase of the RNN model, the sequence of historical data is processed step by step. The RNN takes the current input to update its internal state by incorporating the information from previous time steps through its hidden state. For instance, with a context size of 1, the context size refers to the number of previous timestamps that are considered when making predictions for the next timestamp. At the initial t_0 timestamp, the RNN receives the input value of t_0. It processes this input, updates its internal state, and produces a prediction for the next time step. Moving to the second t_+1, the RNN takes the input value for t_+1 and combines it with the hidden state representing the information from t_1. The RNN then updates its internal state again and generates an output. This process is repeated for each subsequent timestamp, where the RNN considers the current input along with the information stored in the hidden state from previous timestamps. By incorporating the historical context through the hidden state, the RNN is then able to capture patterns and dependencies in the time series (Cho et al., 2014; Elman, 1990). After processing the last timestamp of the input sequence, the RNN has analyzed all the available historical data. It has learned from the patterns and dependencies in the sequence, enabling it to make predictions by leveraging the temporal information and dependencies within the data.

The context size is an adjustable parameter, that specifies n timestamps as context and produces for the next timestamp t_x. For example, a context size of 10 means that the first 10 timestamps are used as the initial context. It processes this context and produces a prediction for the 11th timestamp. As previous studies show (Cho et al., 2014; Elman, 1990), a larger context size can capture longer-term dependencies, but may increase computational complexity. On the other hand, a smaller context size may focus more on recent patterns but may overlook longer-term trends. For spare parts forecasting, a shorter context size can therefore be advantageous. As the demand pattern often exhibits high volatile demand size and irregular intervals between demands. The RNN may then be able to quickly respond to changing demand patterns and capture the dynamics of spare parts demand, thereby improving its forecasting accuracy.

3.7 The LSTM Method

Long-range sequence context preservation is a challenge for RNNs as lengthy sequences cannot be processed by RNN (Cho et al., 2014). As a function of time, the influence of a certain input within the hidden layer decays exponentially, which causes gradients toward zero, preventing the network from learning new weights. To overcome the vanishing gradient problem, LSTM can be the optimal solution (Hochreiter& Schmidhuber, 1997).

LSTM operates in a chain structure in so-called "the cell state", which is a vector transfer running down the whole chain (Sak, Senior, Françoise, 2014). LSTMs are capable of modifying the cell state by removing or adding information, which is carefully controlled via several gates that consist pointwise multiplication process, i.e. vector addition, and a layer of sigmoid neural networks (Elman, 1990).

Selecting information from the cell state to discard is the first stage in LSTM. The forget gate layer, a sigmoid layer generates a number between 0 and 1 in the cell state C_{t-1}at h_{t-1} and x_t, whereas 1 means saving the data, otherwise 0 discarding the data. The first step can be formulated as follow:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right). \tag{8}$$

The next step is choosing which values i_t should be updated by creating a set of new candidate values $\tilde{C_t}$ using the input gate layer via both sigmoid and tanh layer looking at C_{t-1} in h_{t-1} and x_t:

$$\begin{aligned} i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right), \\ \tilde{C_t} &= \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right). \end{aligned} \tag{9}$$

In order to update the C_{t-1} into C_t, f_t is in a linear combination manner with the C_{t-1} in addition with the set of new candidate values:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C_t}. \tag{10}$$

In the final step, the output will be selected and filtered based on the status of the cell. A sigmoid layer to determine which portions of the cell state will be output. Then, we multiply the output of the sigmoid gate by the cell state to ensure that we only output the portions we have selected, after pushing the values to be between 0 and 1 using the tanh function:

$$\begin{aligned} o_t &= \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right), \\ h_t &= o_t \cdot \tanh\left(C_t\right). \end{aligned} \tag{11}$$

3.8 Setting RNN and LSTM

The proposed model for this research is a classic RNN on the work of Elman (1990). It is a multi-Layer Elman RNN with an MLP decoder. First, the data needs to be normalized as having different demand size can impact the forecasting performance, the observations are normalized to a range between 0 and 1. Next, a hyperparameter tuning process is conducted to find the optimal combination of parameters. RNN and LSTM from the Nixtla library (Neuralforecast, 2023) are implemented and the following parameters are hyper-tuned:

1. The size of the context window, which denotes the number of previous time steps considered as input. It is varied within the range of 4 to 12.
2. The encoder amount of layers denotes the number of LSTM or RNN layers in the encoder component of the model, different values ranging from 2 to 6 are explored.
3. The encoder hidden size parameter refers to the number of hidden units in each layer of the encoder. It is varied from 200 to 400.

3.9 The ESRNN Method

The family of capable machine learning models known as the neural network (NN) is diverse. However, they share in common by not being time-specific. It is necessary to preprocess the series data, but the preprocess is difficult as NNs typically contain too many weights to fit for each time series. This problem may be resolved via cross-learning and preprocess (normalization and often deseasonalization), but the results can still be disappointing as NN and even RNN tend to average their responses. Smyl (2018) approaches this problem by being hierarchical–part-time series-specific and part global. It combines manually written components, such as the ES with RNN. Smyl (2018) proves that his approach does not simply combine neural networks and exponential smoothing. Instead, all parameters, including the initial ES seasonality and smoothing coefficients, are fitted simultaneously with the RNN weights using the same gradient descent method.

The model starts with the simplest algorithm of ES:

$$\widehat{y_{t+1}} = \widehat{y_t} + \alpha \left(y_t - \widehat{y_t} \right). \tag{12}$$

Where α is the smoothing factor varying from 0 to 1, according to the algorithm, the forecast is identical to the previous forecast adjusted by the previous error. Smyl (2018) uses an extended version of ES, Holt-Winters, with multiplicative seasonality to observe time series components.

$$\begin{gathered} l_t = \alpha \left(y_t / s_t \right) + (1 - \alpha) \left(l_{t-1} + b_{t-1} \right), \\ s_{t+m} = \gamma \frac{y_t}{(l_t + b_t)} + (1 - \gamma) s_t. \end{gathered} \tag{13}$$

Where α, β, γ are smoothing coefficients between 0 and 1, l denotes level and s denotes multiplicative seasonality. Since ES is fitted for each series, cross-series learning is not possible. However, an NN trained on all series can be used for nonlinear forecasting. The RNN is trained with all the time series by being global, whereas the ES parameters are unique to each time series. The outcome of RNN is used and integrated into the ES model, which has shared parameters to learn the local trends among the series. Therefore, the forecasting becomes:

$$\hat{y}_{t+1..t+h} = RNN \left(X_t \right) * l_t * s_{t+1..t+h}. \tag{14}$$

Where X_t is a vector of preprocessed data and the multiplication is element-wise, $RNN \left(X_t \right)$ denotes the predicted trend by RNN. Moreover, X_t is composed of normalized and deseasonalized time series-derived parameters. The hybrid model optimizes two losses; quantile loss and regularization:

$$L_q(y, \hat{y}) = q(y - \hat{y})_+ + (1 - q)(\hat{y} - y)_+. \tag{15}$$

However, it is possible when treating all series as a single global model might overlook the individual behaviors of each spare part as each of them often exhibits high intermittency, meaning that spare parts have infrequent demand patterns, unique characteristics, and unique demand patterns. Moreover, spare parts demand can be influenced by local factors specific to each part, such as the life cycle of the equipment it belongs to, replacement cycles, and specific customer demands. Training a global model might not capture these local trends and dependencies. In this context, it is still interesting to explore whether a partially and globally trained model like ESRNN can outperform Croston and its variations.

3.9.1 Setting ESRNN

To run the ESRNN model from the Neuralforecast package (Neuralforecast,2023), it is necessary to ensure that the simulated and industrial data sets have the same dimensions as the data in the M4 competition where the model was originally applied. This involved arranging the data such that all time series are consolidated into one column, identified by a unique ID per stock keeping unit (SKU) recorded in a separate column. These included calculating the number of timestamps and unique numbers present in the data. For instance, in the case of the simulated data sets, which initially have dimensions of (60,6500), a transformation is required to restructure the data into a shape of (3,390000), which are unique ID, timestamps 1 to 60, and the value y at timestamp t per SKU, and up to 60 6500 = 390.000 rows.

One particular parameter of interest for potential optimization is the input size, the same as the context size mentioned in RNN and LSTM, which denotes the number of previous time steps considered as input. It is chosen to not adopt any parameters as it requires modifications to the data loading and preprocessing tools within the nixtlats-data-tsdataset-module, which are considered as technical limitations. For this method, a decision is made to keep the input size fixed at 20% of the length per SKU per training data. I.e., the input size is 8 for simulated data, 8 for Oil, 12 for BRAF, 21 for MAN, and 3 for AUTO.

One limitation of the ESRNN model from this package is its inability to run time series with zero demand without disruptions. When a zero demand occurs within a series, the model encounters errors and fails to continue running. To address this issue, a practical solution can be implemented by adding a small bias to each observation within the time series. Specifically, a constant variable of 0.01 is added to all data points. By doing so, even if the demand is zero, the series will have a small non-zero value, allowing the ESRNN model to handle it without encountering errors.

3.10 Equal weighted and Stacking Combination Methods

The practice of combining two or more forecasting methods with the goal of improving overall prediction performance, so called ensemble methods will be described in this section. Several methods will be employed to mitigate the errors associated with individual methods. Two commonly combined techniques will be used. (1) Simple Average: assigning equal weights for each method's predictions. (2) Stacking: use of a Meta-learner to combine predictions from a pool of models (Montero-Manso et al., 2019), a Meta-learner is a model that is used to determine the optimal combination weights for forecast combination.

Montero-Manso et al.(2019) proposed a combined forecasting method by utilizing time series characteristics in two separated processes: to reduce the loss function resulting from a weighted forecast combination, they first employed a collection of time series to train a Meta-model for weighing multiple potential forecasting approaches. Features that are taken from each series served as the Meta-inputs. New series are estimated using a weighted forecast combination in the second phase, where the weights are derived from the previously trained Meta-model. The objective is to find a function optimized by XGBoost that assigns weights to the pool of approaches to minimize the expected loss that will have been produced if the weights are assigned randomly. The Feature-based Forecast Model Averaging (FFORMA) frameworks' operation proposed by Montero-Manso et al.(2019) is split into two phases: (1) A phase when the Meta-learner is trained, and (2) during which the previously trained Meta-learner is used to determine forecast combination weights for a new series, schematically:

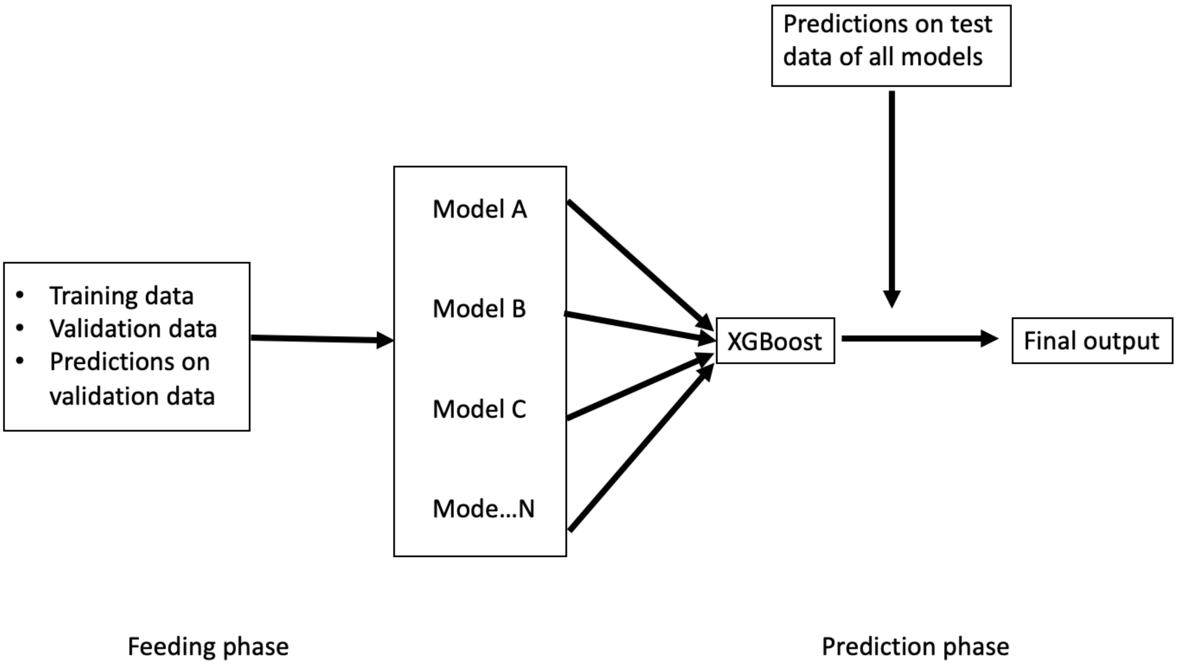

Figure 1: Meta-learning using XGboost. Each base learner is trained and the predictions on the validation, i.e., features serve as Meta-input. During prediction, the Meta-learner updates and combines the predictions on the test set of each base learner.

As Figure 1 shows, the original training data is further divided into training data less the validation data. The validation data is reserved as a target reference in the feeding phase, and the predictions on the validation data are utilized as a feature to train the model.

3.10.1 Setting Equal weighted and Stacking Combination Methods (Meta-learners 1 and 2 and Ensemble 1 and 2)

In the stacking approach, the predictions from all models on the validation set are utilized, and XGBoost is used as an optimizer. The training data, excluding the validation data, is used as a feed feature, while the predictions on the validation data are used as the target. In this case, the XGBoost model is also optimized. Since XGBoost is a regression tree-based model, the depth of the trees will be optimized to achieve the best performance. Once the model has been trained, the predictions on the test data from all the base learners are combined and weighted. Two types of stacking are performed in this research. The first type included only the Croston-based methods, where the predictions from these methods are combined. The second type involved combining the predictions from the Croston-based methods along with RNN. Moreover, to obtain the final prediction for the simple average, i.e., assigning equal weights for each methods, the accumulated predictions on the test data are divided by the number of models.

Therefore, a total of four combined methods are implemented and referred to as ensemble 1, ensemble 2, Meta-learner 1, and Meta-learner 2, where '1' denotes the first pool and '2' denotes the second pool of methods. The term 'ensemble' represents combination methods that are equal-weighted, and the term 'meta_learner' refers to the stacking.

3.11 The ADIDA Method

Nikolopoulos et al. (2011) proposed ADIDA to reduce the number of zero periods by using temporal aggregation, i.e., combining multiple lower levels into higher levels of time series: daily time series into, weekly, monthly, and for long time series into yearly. Nikolopoulos (2011) assumed intermittent times series arrogation, which reduces the amount of zero demand interval, can reduce the noise, and increase forecast accuracy.

At each aggregation level, n observations will be added together backward as a unique block until there are no more observations left. At the stage of predicting, the predicted value is divided by the aggregation factor that denotes the level at which the original time series was aggregated, schematically:

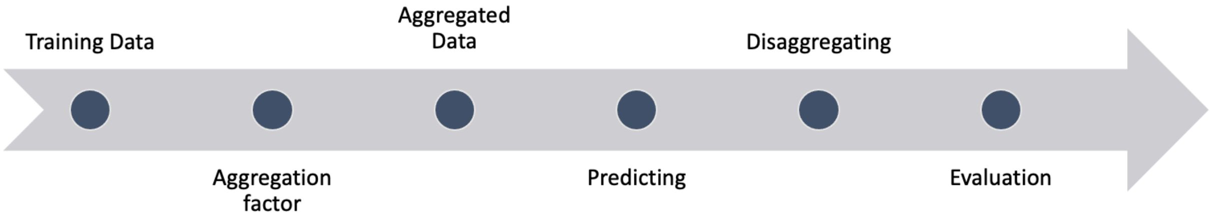

Figure 2: The ADIDA Method Workflow. Aggregating level 1 denotes the original training set, level 2 denotes 2 obsvertaions each block. Each predicted aggregated value will then be disaggregated.

According to Nikolopoulos et al. (2011), when comparing the use of SBA with and without the use of ADIDA, it was found that the results were poorer when SBA was applied to aggregated BRAF data. However, the authors argued that the aggregation process should be conducted carefully to avoid having too few observations, as this could potentially lead to an increase in error.

3.11.1 Setting ADIDA Method

For the simulated data sets (BRAF and OIL), aggregation levels 2 to 6 are selected to analyze the impact of different levels of temporal aggregation. However, due to the shorter timestamps for SKUs in the AUTO data set, aggregation levels 2 to 4 are considered, and in the case of the MAN data set, the aggregation process begins at level 5 and extends up to level 9. In total 39 aggregated data sets are generated. For simplicity and consistency with previous research by Nikolopoulos et al. (2011), only the SBA method is applied to the aggregated data sets. Each aggregated data set is evaluated using the four evaluation measures, and the aggregation level with the best accuracy is selected for further analysis. During the process of aggregating time series, it is common for some observations in the training set to be excluded due to the ratio between the length of time and the chosen aggregation factor for not resulting in whole numbers. In such cases, each decimal place is rounded down to the nearest integer. For example, let's consider the training data from SIM1, which consists of 42 timestamps. If we apply an aggregation level of 2, all observations can be grouped into 21 blocks. Similarly, at aggregation level 3, 14 blocks can be generated. However, at aggregation level 4, only 10 blocks of 4 can be created, resulting in the first and second observations being left unused.

During the process of disaggregation, a similar situation occurs as with the aggregation process. Depending on the chosen aggregation level, a specific number of predictions is made, and each prediction is divided by the aggregation factor to obtain the disaggregated values. For example, let's consider an aggregation level of 2 in simulated data. In this case, 9 predictions will be made. Each of these predictions will be divided by the aggregation factor, resulting in 18 disaggregated predictions for the simulated data. At aggregation level 3, 6 predictions will be estimated, which will lead to 18 disaggregated predictions as well. However, at aggregation level 4, 5 predictions are estimated. After disaggregation, the total number of predictions is then 20, which is greater than the length of the test set. In such cases, it is chosen to remove the last 2 predictions after disaggregation.

4 Data Description

The next step involves data exploring, data cleaning, and generating simulated data sets (Boylan et al., 2008; De Haan, 2019). In this chapter, we discuss data sets to which forecasting methods are applied. In order to ensure reproducibility and validity, two types of data sets are used: industrial data and simulated data sets. Simulated data is generated using the cut-off criteria retrieved from Boylan et al. (2008), four types of demand are simulated: erratic, lumpy, smooth, and intermittent. The objective is to assess whether the findings from the simulated data set align with the empirical data set. Besides generating simulated data, aggregated data in n level for both industrial and simulated are provided. The names of the data sets and the corresponding abbreviation are provided in Table 1.

Table 1: Data Sets

Data.sets	Abbreviation
Automotive Company	AUTO
Manufacturing Firm	MAN
British Royal Air Force	BRAF
Refinery Oil Company	OIL
Simulated 1	SIM1
Simulated 2	SIM2
Simulated 3	SIM3
Simulated 4	SIM4

4.1 Empirical Data

Four industrial data sets are included. To begin, missing values in the data sets have been replaced by zero, and SKUs with zero demand occurrences have been removed. The first data comes from a Dutch manufacturing company consists 3451 SKUs recorded over 150 weeks. The second data set, used by Teunter and Duncan (2009), consists of sales data on 5000 aircraft spare parts from the British Royal Air Force over 84 weeks. The third data set (Syntetos and Boylan, 2005) comes from the automotive industry and comprises sales of 3000 items over two years. The fourth data (Porras and Dekker, 2008) set covers sales data over 56 weeks for 14523 SKUs for an oil refinery from January 1997 to August 2001. These data sets have already undergone cleaning and preprocessing by Nguyen (2023) and are available for use on her GitHub repository.

Descriptive statistics of each industrial data set are provided in Tables 2, 3 ,4, and 5. For every SKU across the data, the mean and the standard deviation (sigma) are provided. The demand size is calculated as the sum of the demand divided by the number of occurrences per time series. For example, the mean minimum for demand size in MAN is 0.084 while the mean maximum, the fourth quantile is 1149.910. Furthermore, the mean inter-demand interval refers to the mean time period between two positive demands per time series, and the mean demand per period denotes the average demand quantity during each time period.

To simplify the analysis, we decide to set the lead-time to zero for every SKU in both the industrial and simulated data sets. Additionally, all exogenous variables are removed such as minimum and maximum order quantity, current stock quantity, and the lead-time per SKU. This is done because not all industrial data sets have the same variables, lead-times, and including different variables would make the results of the applied methods less comparable across the data sets. Moreover, since the price feature has in this analysis an important role in determining inventory performance, pricing data has to be incorporated into both the AUTO and the simulated data sets, as they are originally missing this information. The price data is generated using a framework that relies on the correlation between pricing and its temporal order frequency, as described in De Haan (2021). The pricing information for each data set can be found in Table 13 and Table 14 in the Appendix.

Table 2: Data Description BRAF

Quantile	Demand.per.period	Demand.sizes.	Demand.Intervals
Minimum (std)	0.036 (0.186)	1.000 (0.000)	3.786 (0.000)
1st Quantile (std)	0.155 (0.535)	1.556 (0.745)	7.222 (4.924)
Median (std)	0.369 (1.443)	3.833 (2.871)	9.000 (6.359)
3rd Quantile (std)	1.154 (4.408)	11.333 (8.727)	11.833 (8.074)
Maximum (std)	65.083 (274.062)	668.234 (798.233)	29.523 (14.295)

Table 3: Data Description OIL

Quantile	Demand.per.period	Demand.sizes	Demand.Intervals
Minimum (std)	0.036 (0.187)	1.000 (0.000)	1.000 (0.000)
1st Quantile (std)	0.073 (0.287)	1.000 (0.000)	5.400 (1.641)
Median (std)	0.145 (0.454)	1.500 (0,433)	8.000 (4.000)
3rd Quantile (std)	0.291 (0.913)	2.833 (1.225)	12.000 (6.500)
Maximum (std)	232.727 (599.107)	1600.000 (1523.778)	37.000 (25.000)

Table 4: Data Description MAN

Quantile	Demand.per.period	Demand.sizes	Demand.Intervals
Minimum (std)	0.002 (0.015)	0.084 (0.000)	1.000 (0.000)
1st Quantile (std)	0.2737 (1.275)	3.400 (2.155)	4.028 (2.651)
Median (std)	0.962 (3.788)	8.96054 (7,022)	8.000 (5.377)
3rd Quantile (std)	3.112 (10.714)	22.865 (18.889)	15.000 (9.861)
Maximum (std)	1149.910 (4060.580)	10780.454 (7124.374)	92.000 (48.000)

Table 5: Data Description AUTO

Quantile	Demand.per.period	Demand.sizes	Demand.Intervals
Minimum (std)	0.542 (0.493)	1.000 (0.000)	1.045 (0.208)
1st Quantile (std)	1.458 (1.291)	2.050 (1.108)	1.100 (0.300)
Median (std)	2.333 (1.881)	2.885 (1.716)	1.235 (0.498)
3rd Quantile (std)	4.166 (3.428)	5.000 (3.269)	1.437 (0.718)
Maximum (std)	129.167 (120.162)	193.758 (98.971)	2.091 (1.585)

4.2 Data Classification

In this section, the framework proposed by Boylan et al. (2008) is applied to the classification of each SKU according to its characteristics. Prior studies (Boyland et al., 2008; Kourentzes, 2014) showed that the models' performance can be influenced by the characteristics of each SKU. Boylan et al. (2008) assumed that SBA outperformed the Croston's method and weighted moving average approach (EWMA) in predicting lumpy, erratic, and intermittent demand.

Two cut-off values are used for classification: the mean inter-demand interval, denoted by p, and the squared variation of demand size, denoted by CV^2. The value of p is calculated as the ratio between the sum of T periods and C counts of the non-zero periods. CV^2 represents the ratio between the standard deviation of non-zero demands and the mean of the zero demands. Both cut-off values are used to classify whether demand is lumpy, erratic, smooth, and/or intermittent. The demand data sets are categorized as erratic when $p < 1.32$ and $CV^2 \geq 0.49$), lumpy (when $p \geq 1.32$ and $CV^2 \geq 0.49$), smooth (when $p < 1.32$ and $CV^2 < 0.49$), and intermittent (when $p > 1.32$ and $CV^2 < 0.49$):

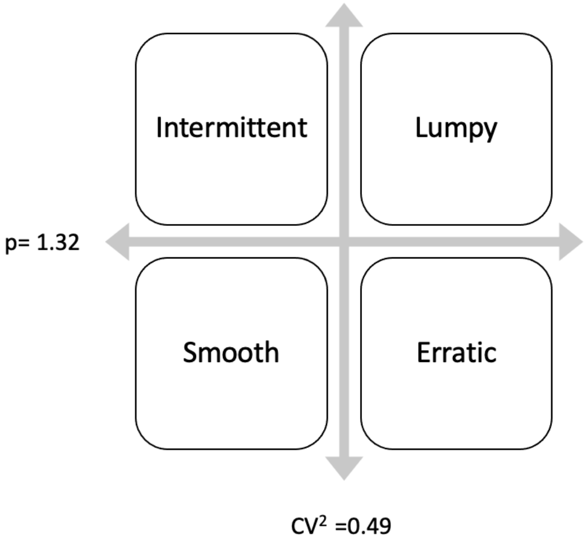

Figure 3: Cut-Off Values. X-as denotes the squared coefficients of varation of zero demand, and y-as the squared variation of demand size.

Based on the cut-off values, the average length of time series, and the average amount of SKUs in industrial data sets. Four simulated, erratic, lumpy, smooth, and intermittent data sets are generated. The simulated data sets were generated using the "tsintermittent" package in R. This package allows us to create data sets by specifying both the parameters p and CV^2. SIM1 represents the erratic pattern with $p = 1$ and $CV^2 = 0.75$, SIM2 corresponds to the lumpy pattern with $p = 1.5$ and $CV^2 = 0.8$, SIM3 represents the smooth pattern with $p = 1.05$ and $CV^2 = 0.3$, and SIM4 represents the intermittent pattern with $p = 1.45$ and $CV^2 = 0.25$. All simulated data sets have the size of 60 timestamps and 6500 SKUs, based on the average timestamps and SKUs of industrial data sets. Table 6 shows the descriptive classification of both industrial and simulated data sets, the industrial data sets have relatively more mixed SKUs while in the simulated data sets is clearly to observe which type of SKU is dominating. Please note that in Table 6, CV^2 is represented as CV2 due to the formatting in Rmarkdown.

Table 6: Demand Description Based CV^2 and p

Data set	CV2	p	Erratic	Intermittent	Lumpy	Smooth
MAN	0.92	16.41	23	562	806	1
BRAF	0.63	11.14	0	2905	2095	0
AUTO	0.41	1.32	378	1074	307	1241
OIL	0.18	14.52	0	6830	814	0
SIM1	0.75	1.00	6198	0	0	302
SIM2	0.80	1.50	410	451	5614	25
SIM3	0.30	1.05	36	0	0	6464
SIM4	0.25	1.45	1	5706	7	786

4.3 Aggregation of Data

In the research conducted by Nikolopoulos et al. (2011) on the BRAF data set, it was observed that the SBA method yielded optimal results at lower frequency levels based on the Mean Absolute Error (MAE). To explore this further, they experimented with different aggregation levels, where higher aggregation levels corresponded to lower frequency levels. The original training data will be aggregated at lower frequency levels, allowing for the generation of aggregated data with varying time intervals. This approach enables the examination of forecasting performance across different aggregation levels. In the current empirical setting, the simulated data, as well as the OIL and BRAF data sets, will undergo a similar aggregation process by aggregating the data from level 2 to level 6. I.e., monthly data sets are transformed into two-monthly data, three-monthly data, and so on. Since the AUTO data set has shorter timestamps for SKUs, aggregation levels from 2 to 4 are considered. Lastly, for the MAN data set, aggregation starts from level 5 and continues up to level 9. In Table 7, the impact of aggregation level 3 on the mean demand size CV^2 and the mean inter-demand interval p for industrial data sets can be observed.

Table 7: Aggregation Level 3

Data set	CV2	p	Erratic	Intermittent	Lumpy	Smooth
MAN	4.35	1	1301	0	0	91
BRAF	4.82	1	5000	0	0	0
AUTO	0.24	1	269	0	0	2731
OIL	4.39	1	7641	0	0	3

As anticipated with the exception of the AUTO data set, the mean inter-demand interval p decreases when demands are aggregated due reducing of zero demand periods. The increase in the mean demand size CV^2 can be explained as demands is larger due to the aggregation.

5 Accuracy Measures

In this research, the performance of different forecasting methods is compared using various performance measures. For intermittent demand forecasting, several accuracy measures are commonly used, including MSE and MASE (Pinçe et al., 2021). MSE is mostly used to assess the individual performance of a forecasting model, while MASE is independent of the scale of the time series by providing a meaningful measure of forecast accuracy relative to a benchmark method: a one-step naïve forecast that assumes the future value will be the same as the most recent observed value (Koehler, 2006).

Moreover, it is important to note that multiple accuracy measures should be used, as each measure has its advantages and drawbacks (Goodwin and Lawton, 1999; Kolassa, 2020; Koutsandreas et al., 2021). In this research, we incorporate the forecasting measures proposed in the M5 forecasting competition, which includes scaled errors that possess appropriate statistical properties. Standard accuracy measures such as MSE and rooted mean squared error (RMSE) are also utilized as they are commonly used in forecasting evaluations.

The mean squared error:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{16}$$

is a commonly used measure that quantifies the average squared difference between the predicted values and the actual values. It provides an overall assessment of the models' accuracy.

The mean absolute scaled error:

$$\text{MASE} = \frac{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}, \tag{17}$$

is the performance measure used in M4 and M5 forecasting, that compares the predicted values to the benchmark model. It considers the scale of the time series and provides a normalized error metric, allowing for comparisons across different time series.

The rooted mean squared error:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{18}$$

Equation (18) is the square root of the MSE and provides a measure of the average magnitude of the forecast errors.

The rooted mean squared scaled error:

$$\text{RMSSE} = \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\frac{1}{n-1}, \sum_{i=2}^{n}(y_i - y_{i-1})^2}}, \tag{19}$$

is a variation of RMSE that incorporates the scaling of the time series by taking the variability of the historical data into account.

In various fields, including time series forecasting and supervised learning, the performance of models is commonly evaluated using metrics such as MSE and RMSE. These metrics assess the average squared difference between predicted and actual values, indicating the overall magnitude of errors. To ensure a more standardized and scaled score when comparing different time series, RMSSE and MASE are commonly used in addition to MSE and RMSE.

6 Inventory Control

Van Wingerden et al. (2014) argue that evaluating forecasting methods based on the accuracy performance may not capture their practical relevance, and follow the methodology outlined by Durlinger and Paul (2012), a base stock level is based on an evaluation of past demand patterns. In this section, an inventory measure outlined by Nguyen (2023) and De Haan (2021) is used. An order-up-to-level policy is implemented to replenish inventory as the stock level falls below the reorder point. This approach aims is to find an optimal quantity of stock on a condition of a certain service level. As mentioned earlier, each SKUs across all data sets are assumed to have a lead-time as 0, indicates immediately stock replenishment. However, it is important to note that even with a lead-time is 0, stockouts are not excluded as uncertain demand still needs to be met.

Next, target fill rate (TFR) will be set to generate the trade-off curves between the TFRs and the AFRs , and between the AFRS and the inventory holding costs. The corresponding stock levels are determined based on the forecasts generated by the methods, along with the variance of the historical demand. These parameters are then used to fit a gamma distribution and compute the base stock levels. The most widely used distribution in demand forecasting is the normal distribution. However, this may not be well-suited for spare parts forecasting, as spare parts are characterized by intermittency, which causes skewness and heavy-tailed distributions due to long periods of zero demand (Moor and Strijbosch, 2002; Nguyen, 2023; Burgin, 1975). According to Nguyen (2023), who applies Croston and its variants, SES, MLP, LightGBM, and Willemain to the same data as in this paper, the achieved fill rates (AFR) are higher, except for Willemain, when the forecasts follow the gamma distribution compared to the normal and negative binomial distribution.

Therefore, the identical inventory measure employed by Nguyen (2023) is utilized and applied to each forecast. In the subsequent section, a concise explanation is provided regarding Nguyen's implementation of inventory control.

6.1 Inventory Setting

Given an item, a TFR during period t, the base stock level $R_{i,r,k}$ is derived from the gamma distribution, predictions, and historical demand. More specifically, the expected shortage per replenishment cycle ESC(R) is used from Burgin (1975), where R denotes the base stock level. Over a base stock level R, we compute the value of ESC:

$$ESC = k/\alpha - R - k/\alpha \cdot F(\alpha R, k+1, 1) + R \cdot F(\alpha R, k, 1). \tag{20}$$

Where k denotes shape and α denotes rate, these parameters are used in the Erlang distribution, which is a special form of the gamma distribution where k is strictly a positive integer. In this analysis, $k = \mu^2/\sigma^2$, where μ is the forecast, and σ^2 is the variance of historical demands. The shape k parameter will not always result in a whole positive number. Therefore, the generalized form of the Erlang distribution, the gamma distribution, will be used to allow k to take on fractional values. The cumulative distribution function (CDF) of the gamma distribution is denoted as F, α denotes rate of the gamma distribution and is computed as $1/\lambda$ and $\lambda = \sigma^2/\mu$ (Burgin, 1975).

Beginning with an initial value of zero, $R_{i,r,k}$ is systematically incremented by whole integer in each loop until the ESC meets the loss target $(1 - TFR) \cdot \mu$. The target fill rate TFR is defined as:

$$TFR = 1 - \frac{ESC}{\mu}. \tag{21}$$

Equation (21) denotes the fraction of demand that can be supplied from the stock on hand and is a widely used performance measure in inventory control. Equation (21) can be rewritten as $(1 - TFR) \cdot \mu = ESC$, implying that an increase in the TFR requires a higher base stock level to meet the loss target. Assuming $\mu = 1$, $\sigma = 3$ and a TFR of 0.85, this results in $k = 0.111$ and $\alpha = 0.111$. The loss target is $(1 - 0.85) \cdot 1 = 0.15$. To equate the loss target of 0.15 with $ESC(R, \alpha, k)$, note R denotes base stock level, equation (21) suggests that the base stock level should be set to at least $R = 11$, this results in $ESC = 0.126 <$ loss target. ESCs are not always equal to the TFRs due to the policy of increasing the base stock level only in integers. If the TFR increases to 0.90, then the minimum value of R is 13, which results in $ESC = 0.093$, falling below the loss target. The implementation of inventory for a period t in the forecasting horizon F is as follows:

(1) At the beginning of a period, the base stock level $R_{i,r,k}$ is computed and derived from the gamma distribution and the target fill rate (TFR).

(2) The $R_{i,r,k}$ is compared with the actual inventory level, that is the actual inventory level at the end of previous period $IL_{i,r,k-1}$, to determine the replenishment $Q_{i,r,k}$ that arrives during the current period. Therefore, the actual inventory level $IL'_{i,r,k}$ after the replenishment is computed with the $Q_{i,r,k}$ during the period, i.e., $IL'_{i,r,k} = IL_{i,r,k-1} + Q_{i,r,k}$.

(3) For calculation purposes, a separate variable amount of supplied item $S_{i,r,k}$ during a demand period is defined. I.e., the $S_{i,r,k}$ denotes the minimum quantity of stock required to meet the actual demand. The inventory level at the end of a period $IL_{i,r,k}$ is computed as $IL'_{i,r,k} - S_{i,r,k}$.

Furthermore, Nguyen (2023) assumes there are no back orders or initial inventory levels at the beginning of each test set. Moreover, $R_{i,r,k}$ is zero when the forecast $\mu \leq 0$ as the shape parameter k of the gamma distribution requires positive μ value, the part fill rate is also zero when there is no demand in test periods, and $R_{i,r,k}$ is capped to the highest observed demand in training data to avoid extremely high holding costs due to overstocking. However, this approach can also result in understocking for items with demand that occurs in later periods, particularly in cases where there are zero or very few demands observed in the training data, but a high occurrence of demands in the test data. After obtaining $S_{i,r,k}$ for each forecasting period, the achieved part fill rate (APFR) is computed, please note F denotes now the forecasting horizon:

$$APFR_{i,r} = \frac{\sum_{k \in F} S_{i,r,k}}{\sum_{k \in F} D_{i,r,k}}. \tag{22}$$

Consequently, to compute the achieved fill rate of a data set (average-AFR), we average $APFR_{i,r}$ for every item by dividing number of N items:

$$\text{avgAFR}_r = \frac{1}{|I|} \sum_{i \in I} APFR_{i,r} \tag{23}$$

Moreover, taking the ratio of the sum of all $S_{i,r,k}$ and all the occurred demand to obtain the total achieved fill rate (total-AFR):

$$\text{totalAFR}_r = \frac{\sum_{i \in I, k \in F} S_{i,r,k}}{\sum_{i \in I, k \in F} D_{i,r,k}}. \tag{24}$$

In simpler terms, average-AFR denotes the average percentage of demand fulfilled for all items. This means that the low-demand items with a low APFR can significantly decrease the overall average, even if other items have a high APFR. On the other hand, total-AFR calculates the overall percentage of total demand met for all items, regardless of their individual APFR. This suggests that the low-demand items have a smaller impact on the total-AFR because of the low contribution to the overall demand and a high-demand item can significantly affect and lower the total-AFR when the demand is not fulfilled. Therefore, by considering both average- and total-AFR, we can evaluate the models' performance at both the individual item level and the overall level of demand fulfillment.

The evaluation includes the measurement of the AFR in terms of average-AFR and total-AFR. Simultaneously, it involves an analysis of the trade-off between the AFRs and the associated holding costs, aiming to assess the financial implications of various forecasting methods. The holding costs are defined as 25 percent of the item price and the TFRs varied from 0.75 till 0.99.

7 Results

In this section, the results obtained from the mentioned models applied to simulated and industrial data sets will be presented. The first part focuses on forecasting accuracy, which is evaluated using MSE, RMSE, MASE, and RMSSE. Additionally, the results from the ADIDA will be presented in a separate table to highlight the developments of the accuracy performance of each aggregation level. In the second part, we will present the inventory performance of each model on each data set. For illustrative and comprehensive representation, we have decided to present only the RMSSE for all methods applied to eight data sets in Table 8. For the overall performance analysis, please refer to Table 19 in the Appendix.

This research did not actively monitor the runtime per model. Nevertheless, based on estimates, Willemain is presumed to require the longest processing time, approximately 90 to 150 minutes per data set. In contrast, ESRNN, RNN, and LSTM have an average runtime of around 20 minutes per data set. Croston and its variations require about 3 to 5 minutes for each data set. Ensemble 1 and ensemble 2 are executed almost instantaneously once the individual methods have been applied. Meta-learner 1 and 2, on the other hand, typically require an average of 10 to 15 minutes per data set. Lastly, it should be noted that Willemain is executed in R, while the other models are implemented in Python.

7.1 Overall Accuracy Performance

Table 8 shows the RMSSE of the models applied to eight different data sets. A lower error indicates better forecasting accuracy. The performance of the other accuracy metrics is provided in the Appendix, Table 19.

Table 8: Forecasting Performance - RMSSE

Model	AUTO	OIL	MAN	BRAF	SIM1	SIM2	SIM3	SIM4
Croston	1.703	10.923	11.439	5.020	1.008	1.049	0.940	0.921
Croston_T	1.699	10.853	11.439	5.019	1.008	1.049	0.940	0.922
TSB	1.725	12.996	11.437	5.062	1.007	1.048	0.938	0.920
SBA	1.698	10.931	11.428	5.018	1.009	1.050	0.941	0.922
RNN	2.153	10.700	11.995	4.988	1.033	1.114	0.939	0.933
LSTM	2.148	10.570	11.988	4.988	1.035	1.108	0.939	0.937
ESRNN	2.250	10.311	11.963	4.988	1.125	1.258	1.063	1.138
Willemain	1.690	9.877	11.475	4.953	1.001	1.049	0.937	0.926
Ensemble_1	1.705	11.299	11.435	5.029	1.008	1.049	0.940	0.921
Ensemble_2	1.701	10.820	11.485	4.970	1.005	1.050	0.934	0.917
Meta-learner_1	1.750	9.982	11.849	4.916	0.982	1.031	0.912	0.903
Meta-learner_2	1.750	9.982	11.849	4.916	0.982	1.031	0.912	0.904

7.1.1 Findings Simulated Data Sets

Regarding the simulated data sets, Meta-learner 1 and 2 are the best-performing models that are evaluated on MSE, RMSE, and RMSSE. When MASE is selected, Meta-learners 1 and 2 only outperform other methods in SIM3 and SIM4, while RNN and LSTM have the lowest MASE in SIM1 and SIM2. ESRNN is the worst performer among all the models.

Furthermore, ensemble 2 outperforms ensemble 1 across all accuracy measures. This suggests that the inclusion of the RNN model contributes positively to the accuracy performance. However, when compared to stacking methods (Meta-learners 1 and 2), the contribution of the RNN model does not improve the overall performance. The difference between the performance accuracy of both two Meta-learners is negligible, making it difficult to distinguish between them.

7.1.2 Findings Industrial Data Sets

In the case of the BRAF data set, Meta-learner 1 and 2 demonstrate the best performance among the evaluated models. However, surprisingly, ensemble 1 performs the worst across all accuracy measures, which may have been unexpected considering its moderate performance score in simulated data sets. Moving on to the methods applied to the MAN data set, SBA appears to have the best accuracy performance in terms of MSE, RMSE, and RMSSE. ESRNN achieves the lowest MASE score. As for the OIL and AUTO data sets, Willemain has the best performance when evaluated on MSE, RMSE, and RMSSE. ESRNN achieves the lowest MASE score in predicting the OIL data, while ensemble 2 achieves the lowest MASE score in the AUTO data.

7.1.3 Overall Findings

It appears that the stacking method generally exhibits the best performance in terms of MSE, RMSE, and RMSSE in simulated data sets. However, this is not always the case in industrial data sets. In data sets such as AUTO and MAN, the stacking method performs the worst. This can be attributed to the extremely high mean inter-demand interval and demand sizes of MAN data, which could affect the performance of Meta-learners. In the case of AUTO data, the intermittency and demand sizes are comparable to the simulated data, which suggest that the difference in performance can be related to the relatively small timestamp of AUTO. Willemain demonstrates superiority in MSE, RMSE, and RMSSE in the AUTO and OIL data. However, Willemain does not have the same superiority in other simulated or industrial data sets.

Table 9 illustrates whether the RNN, LSTM, ESRNN, Willemain, equal-weighted combined methods, and stacking methods improved the accuracy over the individual performance of the Croston and its variations. Improvement (+) is observed when a model outperforms the individual performance of Croston, Croston optimized, TSB, and SBA. For example, if a model's MSE is lower than Croston, Croston optimized, and TSB but higher than SBA, it is considered as a deterioration (-).

Table 9: Performance Comparsion on Croston Based Methods

Model	Metrics	AUTO	OIL	MAN	BRAF	SIM1	SIM2	SIM3	SIM4	Improvement chance
RNN	MSE	-	+	-	+	-	-	-	-	0.25
	RMSE	-	+	-	+	-	-	-	-	0.25
	MASE	-	-	+	+	+	+	+	-	0.63
	RMSSE	-	+	-	+	-	-	-	-	0.25
LSTM	MSE	-	+	-	+	-	-	-	-	0.25
	RMSE	-	+	-	+	-	-	-	-	0.25
	MASE	-	-	+	+	+	+	+	-	0.62
	RMSSE	-	+	-	+	-	-	-	-	0.25
ESRNN	MSE	-	+	-	+	-	-	-	-	0.25
	RMSE	-	+	-	+	-	-	-	-	0.25
	MASE	-	+	+	+	-	-	-	-	0.38
	RMSSE	-	+	-	+	-	-	-	-	0.25
Willemain	MSE	+	+	-	+	+	-	+	-	0.62
	RMSE	+	+	-	+	+	-	+	-	0.62
	MASE	-	-	-	+	-	-	-	-	0.12
	RMSSE	+	+	-	+	+	-	+	-	0.62
Ensemble_1	MSE	-	-	-	-	-	-	-	-	0.00
	RMSE	-	-	-	-	-	-	-	-	0.00
	MASE	-	-	-	-	-	-	-	-	0.00
	RMSSE	-	-	-	-	-	-	-	-	0.00
Ensemble_2	MSE	-	+	-	+	+	-	+	+	0.62
	RMSE	+	-	-	+	+	-	+	+	0.62
	MASE	+	-	-	+	+	+	+	+	0.75
	RMSSE	-	+	-	+	+	-	+	+	0.62
Meta-learner_1	MSE	-	+	-	+	+	+	+	+	0.75
	RMSE	-	+	-	+	+	+	+	+	0.75
	MASE	-	+	-	+	-	-	+	+	0.50
	RMSSE	-	+	-	+	+	+	+	+	0.62
Meta-learner_2	MSE	-	+	-	+	+	+	+	+	0.62
	RMSE	-	+	-	+	+	+	+	+	0.62
	MASE	-	+	-	+	-	-	+	+	0.50
	RMSSE	-	+	-	+	+	+	+	+	0.62

According to Table 9, Meta-learner 1 has the highest likelihood of improvement over Croston-based methods, while ensemble 1 has the lowest likelihood of improvement. Furthermore, the overall likelihood of improving the MSE, RMSE, MASE, and RMSSE can be determined by calculating column-wise, i.e., for each data set across all methods. For AUTO, OIL, MAN, BRAF, SIM1, SIM2, SIM3, and SIM4, the overall improvement likelihoods are as follows: 0.155, 0.696, 0.091, 0.878, 0.484, 0.273, 0.513, and 0.364. For example, when using stacking or equal-weighted, RNN, and Willemain applied on AUTO data, there is a 15.5% chance of improving one of the four mentioned metrics over Croston, Croston optimized, TSB, and SBA.

Moreover, to determine the model with the best performance across all data sets, a cumulative voting decision has been employed. Each accuracy measure per data set can contribute to a voting score, resulting in a total of 32 votes as there are eight data sets and four metrics.

Furthermore, when the performance of Meta-learner 1 and Meta-learner 2 is equal, it is preferable to choose Meta-learner 1 due to its requirement of fewer features to achieve the same level of performance:

Table 10: Cumulative Voting. Models with a voting score of 0 are not included.

Model	Voting
Meta-learner 1	14
Willemain	6
Meta-learner 2	3
ESRRN	3
SBA	3
RNN	2
Ensemble_2	1

Finding 1. Based on the results presented in Tables 9 and 10, the stacking model Meta-learner 1, demonstrates the highest sum of the likelihood of improvement in various metrics, received the highest number of votes, as indicated in Table 10. This stacking method exhibits the best overall accuracy performance, particularly when the demand is strictly characterized as erratic, lumpy, smooth, or intermittent, as observed in the simulated data. In industrial data, such as the AUTO, OIL, and MAN data sets, the superiority of Meta-learner 1 and 2 has diminished compared to other methods.

Finding 2. In SIM4, where the demand is characterized as intermittent. Three combination methods, both Meta-learners and ensemble 2 show better accuracy performance than Croston and its variations.

Finding 3. The difference between the models' performance in industrial data is larger compared to simulated data. On simulated data, the models show comparable evaluations across all four metrics, except for ESRNN. However, on industrial data such as MAN, OIL, and AUTO, the differences between the models are significantly greater. For instance, the difference in MSE between Croston and RNN on AUTO data is as high as 60%, whereas the difference between these two methods on simulated data ranges between 1% and 12%.

Moreover, when considering scaled error metrics like RMSSE, it can be observed that the difference between the predicted scaled errors among the industrial data sets is much larger than that of the simulated data. For instance, the RMSSE of all models from simulated data sets varies between 0.903 and 1.258, whereas the differences among the industrial data sets are more significant. This indicates that the characteristics of simulated data do not align with those of industrial data. These industrial data sets exhibit much higher intermittency and demand size compared to the cut-off values used to generate the four simulated data sets. Consequently, this leads to a limitation of this research, as the simulated data sets lack the same characteristics as the industrial data sets. Future research should consider generating simulated data with higher intermittency and demand sizes to address this disparity.

Finding 4. In cases when the intermittency is extremely high combined with a high mean demand size such as in MAN data, no single applied model can overall outperform Croston, Croston optimized, TSB, and SBA. Only RNN, LSTM , and ESRNN have a lower error when compared to MASE metric.

Finding 5. Although ensemble 2 shows a similar likelihood improvement over the Croston and its variations as Meta-learner 1, the absolute difference between performance across all metrics in every data sets indicates that Meta-Learner 1 performs better with lower error.

Finding 6. In the AUTO, where the mean inter-demand interval p is 1.32 and the mean demand size CV^2 is 0.41, there is a slight difference compared to SIM4, which has values of 1.45 and 0.25, and SIM3, which has values of 1.05 and 0.30. Despite stacking methods performing exceptionally well in SIM3 and SIM4 and outperforming Croston, Croston optimized, TSB, and SBA. Unexpectedly, Meta-learner 1 and 2 perform worse when applied to the AUTO data. Only Willemain and ensemble 2 show better performance. Compared to the simulated data setting, the cause for this outcome can be that the number of recorded periods in the AUTO is much shorter compared to the simulated data, with 24 periods compared to 60 in the simulated data.

7.1.4 Data Aggregation with SBA (ADIDA)

As aforementioned, the same approach to Nikolopoulos et al. (2011) is adopted, utilizing SBA on aggregated data. According to Table 11, the predictions using aggregated data have shown improvements, especially in the industrial data sets. Significant improvements are observed in the

OIL, BRAF, and MAN data sets. For instance, the performance of SBA in predicting OIL is optimal at aggregate level 4, where the original MSE of 170.916 has been reduced to 139.412. This improvement is also evident in other accuracy measures. In the case of the MAN data set, the optimal aggregation level is 5, beyond which the error starts increasing. On the other hand, the improvement is minimal in the AUTO data. Regarding the simulated data, the predictions have become worse due to aggregation.

When comparing the improvement in accuracy performance with the degree of intermittency, it is observed that the OIL data set shows a higher improvement compared to MAN, despite MAN having a higher level of intermittency. This difference can be attributed to the higher mean demand size in MAN. The demand pattern, including the occurrence and timing of positive demand, may also impact the forecasting accuracy. For instance, in the OIL data, prolonged periods of non-demand often follow a positive demand, potentially indicating patterns like periodic maintenance. However, such patterns may not be as common in other industrial data sets. It is important to note that these factors are beyond the scope of this research and will be acknowledged as a limitation.

Therefore, for the inventory performance, the aggregated level 6 of the OIL data set, denoted as SBA-OIL-6, the aggregated level 5 of the MAN data set, denoted as SBA-MAN-5, and the aggregated level 6 of the BRAF data set, denoted as SBA-BRAF-6 are evaluated. The remaining data sets remain the same.

Finding 7. The predictions using aggregated data have shown improvements, especially in industrial data sets. Significant improvement in accuracy have been observed in the OIL, BRAF, and MAN data sets. However, no improvements have been observed in the AUTO and the simulated data sets. Moreover, it can be inferred that aggregation, similar to lead-time, has an effect on the inventory control assessment. This is because a longer lead-time requires a higher amount of base stock level to meet the demand during that period. A high-demand item characterized by low demand occurrences is also balanced, contributing to a more stable pattern. Therefore, both aggregation and lead-time are expected to have a comparable impact on the need for maintaining inventory levels. However, due to the unavailability of lead-time information, lead-time for each data is set to zero. Future research is needed to determine whether aggregation remains necessary to improve the forecasting performance after incorporating real lead-time information.

Table 11: Aggregating Results of SBA. The first-row block is the original prediction of SBA without any aggregation. Ignore the NAs due to the different starting and ending levels of aggregation for the MAN and AUTO data.

Model	Metric	AUTO	OIL	MAN	BRAF	SIM1	SIM2	SIM3	SIM4
SBA	MSE	85.607	170.916	12958.075	205.273	79.538	78.757	35.295	40.463
	RMSE	9.252	13.073	113.834	14.327	8.918	8.875	5.941	6.361
	MASE	0.608	1.029	0.891	0.922	0.731	0.757	0.719	0.760
	RMSSE	1.698	10.931	11.428	5.018	1.009	1.050	0.941	0.922
Aggregating 2	MSE	87.021	141.996	NA	200.006	78.076	78.353	34.498	40.194
	RMSE	9.329	11.916	NA	14.142	8.836	8.852	5.873	6.340
	MASE	0.619	0.958	NA	0.905	0.729	0.760	0.713	0.758
	RMSSE	1.712	9.964	NA	4.954	0.999	1.047	0.931	0.919
Aggregating 3	MSE	86.667	141.528	NA	200.334	78.275	78.613	34.572	40.313
	RMSE	9.310	11.897	NA	14.154	8.847	8.866	5.880	6.349
	MASE	0.616	0.967	NA	0.905	0.732	0.764	0.716	0.759
	RMSSE	1.709	9.947	NA	4.958	1.001	1.049	0.932	0.921
Aggregating 4	MSE	85.488	139.412	NA	200.835	78.990	79.397	34.837	40.643
	RMSE	9.246	11.807	NA	14.172	8.888	8.911	5.902	6.375
	MASE	0.615	0.960	NA	0.902	0.736	0.768	0.720	0.762
	RMSSE	1.697	9.873	NA	4.964	1.005	1.054	0.935	0.924
Aggregating 5	MSE	NA	141.146	12880.216	200.574	79.358	79.782	35.007	40.815
	RMSE	NA	11.880	113.491	14.162	8.908	8.932	5.917	6.389
	MASE	NA	0.952	0.872	0.897	0.739	0.769	0.722	0.764
	RMSSE	NA	9.934	11.394	4.961	1.008	1.057	0.937	0.926
Aggregating 6	MSE	NA	139.430	12925.675	199.924	79.315	79.724	34.975	40.805
	RMSE	NA	11.808	113.691	14.139	8.906	8.929	5.914	6.388
	MASE	NA	0.957	0.871	0.893	0.739	0.769	0.721	0.764
	RMSSE	NA	9.873	11.414	4.953	1.007	1.056	0.937	0.926
Aggregating 7	MSE	NA	NA	12889.818	NA	NA	NA	NA	NA
	RMSE	NA	NA	113.533	NA	NA	NA	NA	NA
	MASE	NA	NA	0.870	NA	NA	NA	NA	NA
	RMSSE	NA	NA	11.398	NA	NA	NA	NA	NA
Aggregating 8	MSE	NA	NA	12901.995	NA	NA	NA	NA	NA
	RMSE	NA	NA	113.587	NA	NA	NA	NA	NA
	MASE	NA	NA	0.872	NA	NA	NA	NA	NA
	RMSSE	NA	NA	11.403	NA	NA	NA	NA	NA
Aggregating 9	MSE	NA	NA	12926.229	NA	NA	NA	NA	NA
	RMSE	NA	NA	113.694	NA	NA	NA	NA	NA
	MASE	NA	NA	0.872	NA	NA	NA	NA	NA
	RMSSE	NA	NA	11.414	NA	NA	NA	NA	NA

7.2 Inventory Performance

In this section, the term achieved fill rate (AFR) is frequently used as an inventory performance measure. When used without specification, the AFR generally refers to both average-achieved fill rate (average-AFR) and total-achieved fill rate (total-AFR), as the observed pattern is consistent for each target fill rate (TFR) and with inventory holding costs. However, if there is a need to distinguish between the two measures, both average-AFR and total-AFR will be explicitly mentioned.

Furthermore, the goal is to achieve an AFR that matches the TFR. It is essential to avoid overestimating the AFR ($AFR > TFR$) because doing so would lead to holding more inventory than required, resulting in higher inventory costs.

7.2.1 SIM1 Performance

The SIM1 data shows the presence of irregular items, characterized by highly volatile demand sizes with low intermittency. According to Figures 4a and 4c, ESRNN consistently achieves higher average-AFRs and total-AFRs compared to other methods up to the TFR of 0.98. However, the difference in AFR also leads to an increase in holding costs. ESRNN demonstrates the poorest performance among all methods when balancing AFRs and holding costs, whereas Willemain outperforms other models up to a TFR of 0.83, with no significant difference in incurred costs compared to the other models.

Regarding accuracy performance, the stacking methods exhibit superior results, surpassing other methods in accuracy metrics. Willemain closely follows as the second-best performer. In terms of inventory performance, the top-performing methods in accuracy exhibit comparable inventory performance with methods that have lower accuracy performance, such as Croston and its variations.

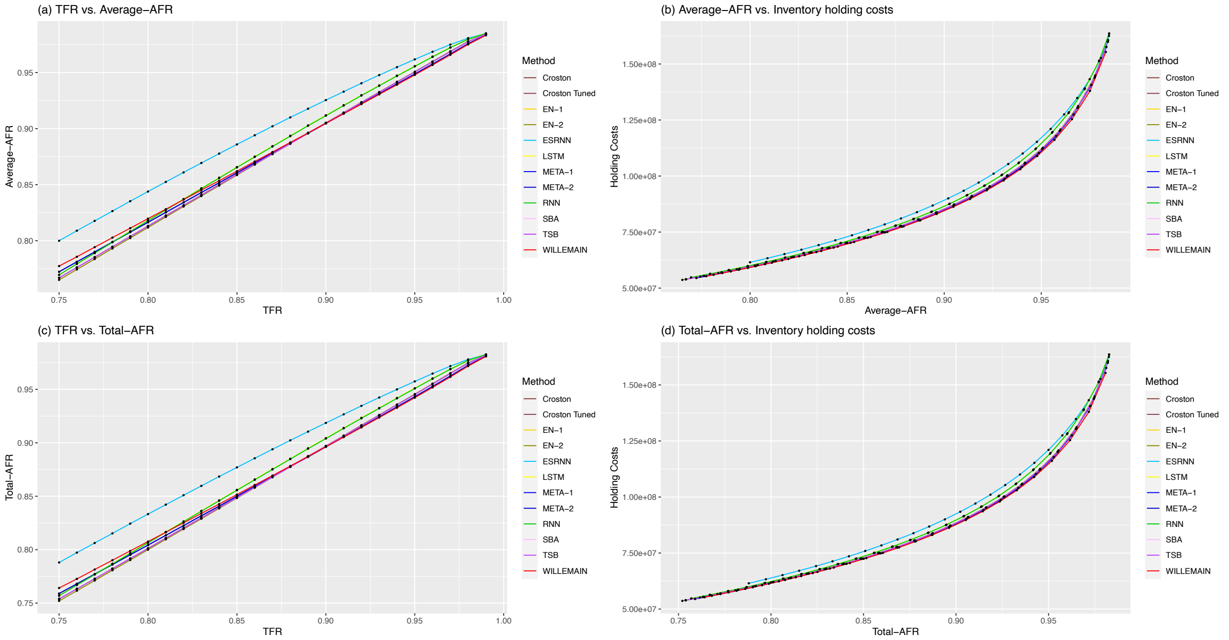

Figure 4: SIM1 Inventory Performance

7.2.2 SIM2 Performance

The SIM2 data is characterized as lumpy, consisting of intermittent and erratic items with a longer mean interval between demands compared to the SIM1 data. Similar to the findings in SIM1, ESRNN achieves higher average-AFRs and total-AFRs at each TFR compared to other models. However, in the trade-off between AFRs and inventory holding costs, as shown in Figure 5b and Figure 5, ESRNN shows the worst performance followed by RNN and LSTM, while the remaining methods perform better with lower inventory costs. On both average-AFR and total-AFR, the inventory holding costs for ESRNN, LSTM, and RNN are higher than all other models until the TFR of 0.99 is reached. Ensemble 1 stands out as the superior model, achieving a higher AFR at every TFR without incurring any further costs compared to other models.

Contrary to the accuracy performance, the superior accuracy performance of Meta-learner 1 and 2 does not seem to correlate with inventory control measures. Both methods demonstrate similar inventory performance to other methods except for ESRNN, RNN, and LSTM. This finding shows, along with the results from SIM1, suggests that evaluating models based solely on accuracy is not sufficient in practice. The findings from SIM1 and SIM2 indicate that higher accuracy does not always translate to a better inventory performance.

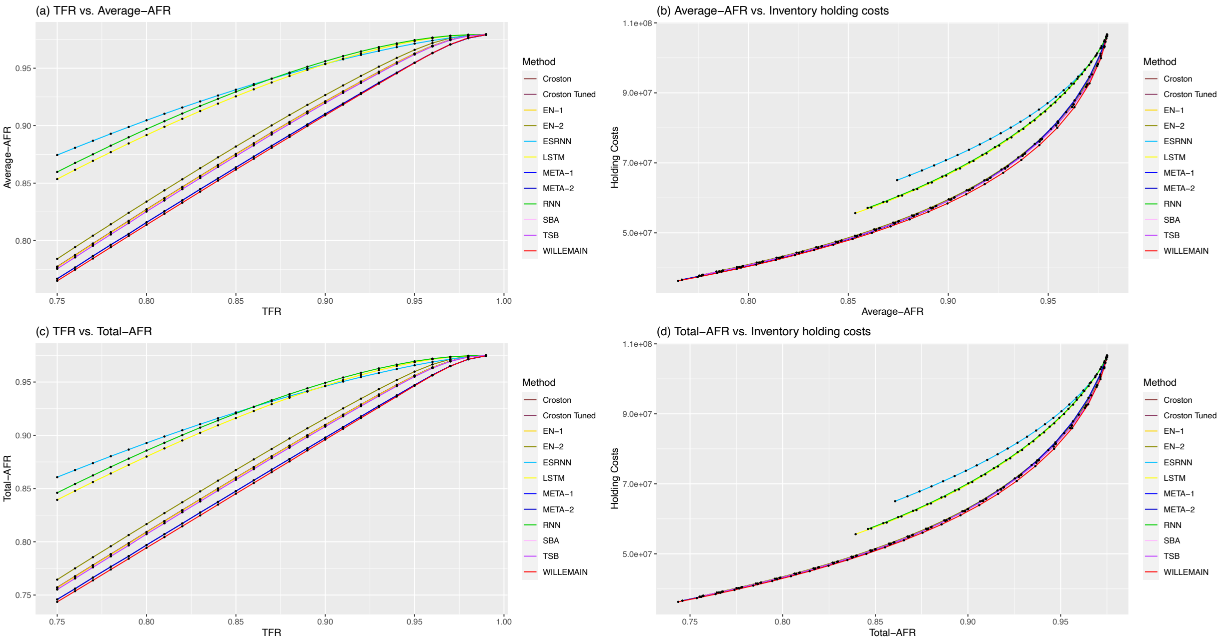

Figure 5: SIM2 Inventory Performance

7.2.3 SIM3 Performance

The SIM3 data is characterized by a smooth demand pattern, with minimal variability in demand size and a low average demand interval. In contrast to the previous findings from SIM1 and SIM2, as shown in Figure 6a and 6c, Willemain emerges as the top-performing model in the trade-off between AFRs and TFRs. Willemain remains superior until a TFR of 0.9, after which the difference becomes less visible and ESRNN surpasses it at TFR of 0.93. In addition, as observed in Figures 6b and 6d, the increase in AFR does not lead to a significant increase in costs of Willemain compared to other models.

Regarding accuracy performance, despite the superior performance of Meta-learner 1 and 2, this has not resulted in a higher inventory performance. Willemain ranked as the third best in terms of MSE, yields the highest AFR at each TFR, while the costs remain the same. ESRNN continues to exhibit poor inventory performance, incurring higher costs at each AFR up to 0.99. This aligns with the previous findings in SIM1 and SIM2.

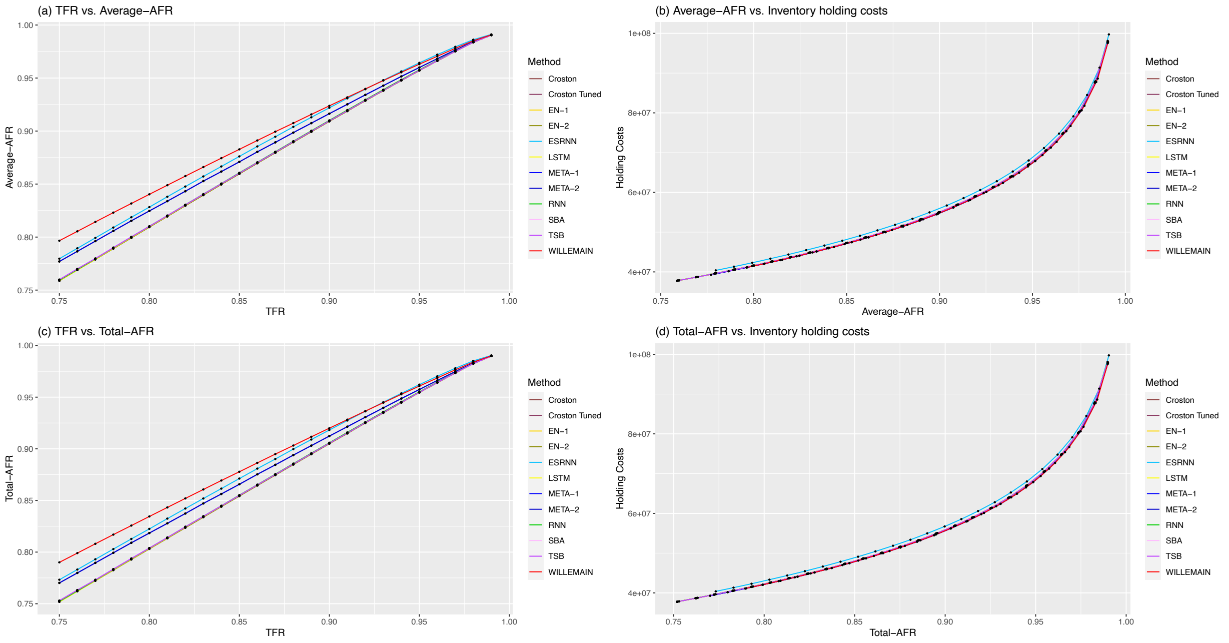

Figure 6: SIM3 Inventory Performance

7.2.4 SIM4 Performance

The analysis of SIM4 data, primarily composed of intermittent items, further supports the consistent findings observed in SIM1, SIM2, and SIM3. Figures 7a and 7c demonstrate that the ESRNN consistently achieves higher AFRs at each TFR, but with the worst performance in the trade-off between AFRs and inventory holding costs. More specifically, LSTM exhibits the worst inventory performance after reaching a TFR of 0.96 in both trade-off curves. Willemain is considered superior due to achieving a higher AFR up to a TFR of 0.85, with inventory holding costs that do not differ from other models. After that point, models, except for ESRNN, LSTM, and RNN, exhibit comparable inventory performance. Furthermore, in this data set, Meta-learner 1 and 2 demonstrate a similar inventory performance to models with lower accuracy, such as Croston and its variations, which further indicates that accuracy does not necessarily correlate with inventory performance.

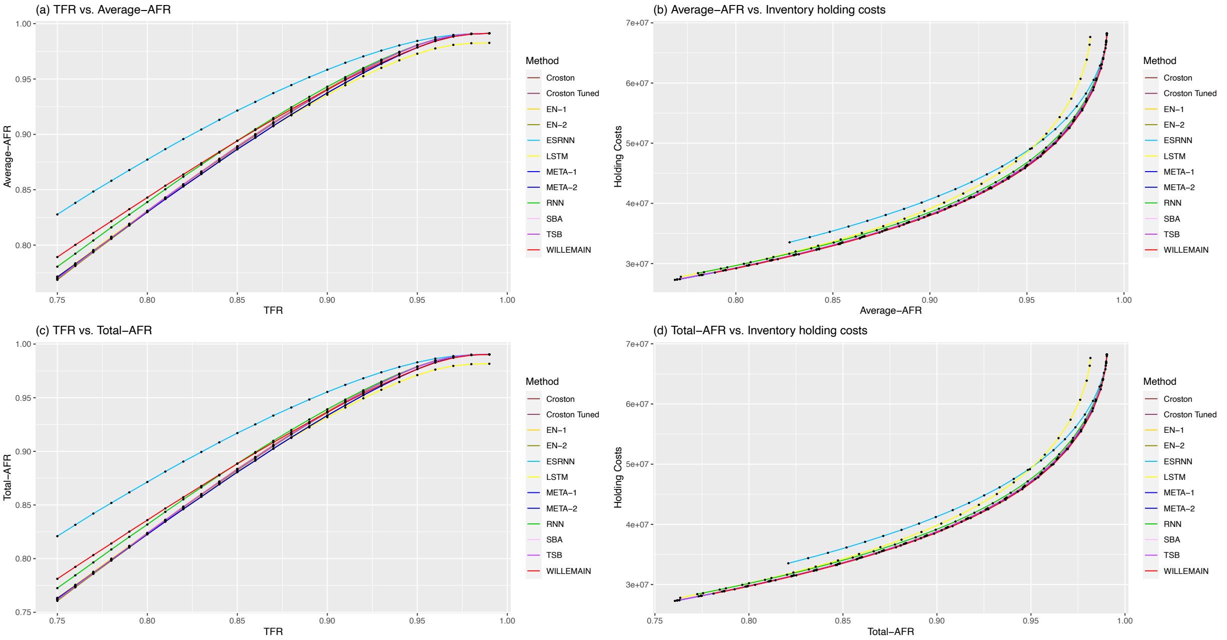

Figure 7: SIM4 Inventory Performance

7.2.5 AUTO Inventory Performance

The AUTO data is characterized by mixed demand patterns, including various types of demand such as smooth and intermittent items. Analyzing Figures 8a and 8c, it becomes clear that ESRNN consistently achieves the highest AFR at each TFR. Figure 8b shows, Willemain is superior in the trade-off between average-AFRs and inventory holding costs, while ESRNN, LSTM, and RNN have the worst performance. However, according to Figure 8d, considering the total-AFR, the costs of ESRNN, LSTM, and RNN are lower than Willemain and other models. This suggests that the Willemain performs better in predicting expensive items with low demand, while ESRNN, LSTM, and RNN excel in forecasting cheaper items with higher demand. Another observation is that LSTM performs worse in the trade-off between total-AFRs and inventory holding costs after reaching a total-AFR of 0.925.

Regarding accuracy performance, Willemain demonstrates superior performance in terms of MSE, RMSE, and RMSSE compared to other models. However, this superiority is only reflected in the trade-off between average-AFRS and inventory holding costs.

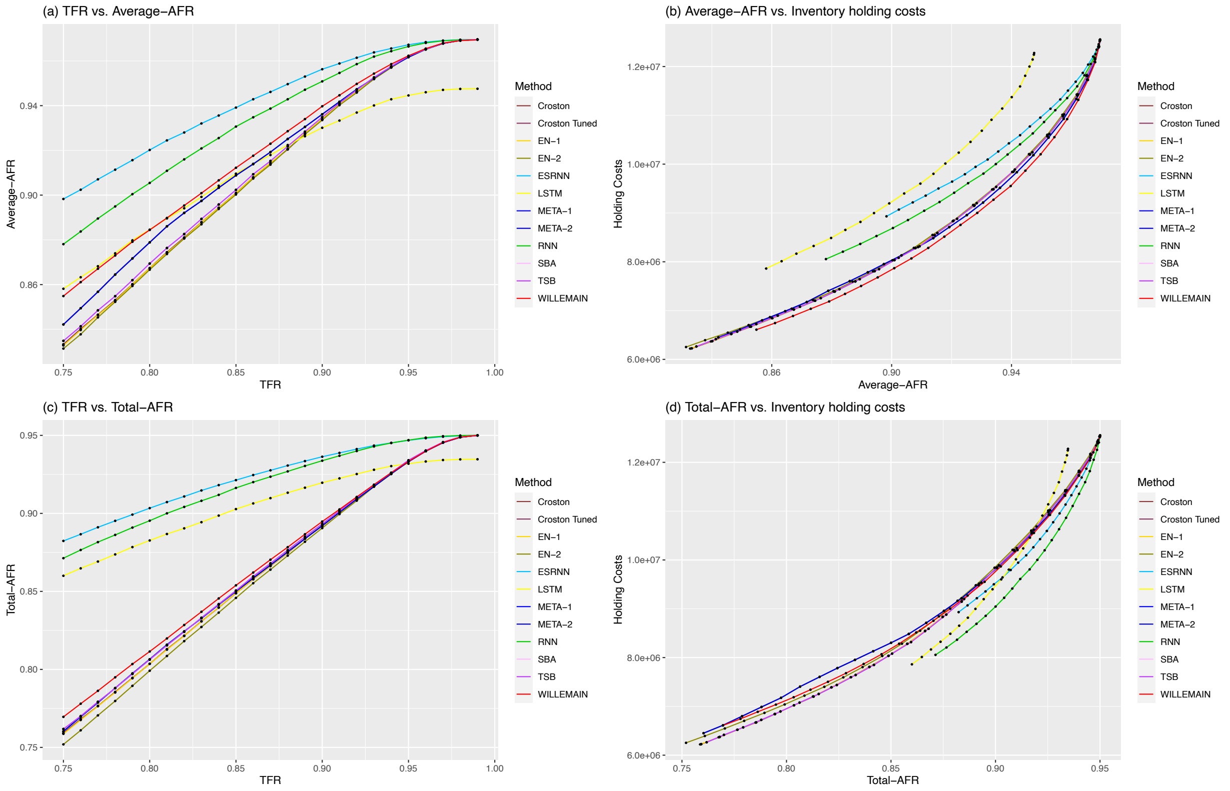

Figure 8: AUTO Inventory Performance

7.2.6 OIL Inventory Performance

In the OIL data, where most items are classified as intermittent and some as lumpy, all models achieve a relatively low average-AFR compared to previous results. With the exception of RNN, all models fall within the range of 0.42 to 0.44 for average-AFR. When considering the trade-off curves between average-AFRs and TFRs, as well as total-AFRs and TFRs, ESRNN and LSTM achieve higher AFRs at each TFR. However, in the trade-off between AFRs and inventory holding costs, they are inferior compared to other models. TSB is superior in the trade-off between average-AFRs and inventory holding costs, while both Meta-learners outperform others in the trade-off between total-AFRs and inventory holding costs. This suggests that the TSB performs better in predicting expensive items with low demand, while the Meta-learners excel in forecasting cheaper items with higher demand. As for RNN, in this case, it consistently predicts a value of 0 for both average- and total-AFR at all TFRs, suggesting that the base stock level is always predicted as 0. This also explains why a black dot is observed at the origin of the graph, indicating no inventory costs incurred due to the inventory performance of RNN.

In contrast to the findings of Nguyen (2023), both the trade-off curves between TFRs and AFRs have shown a decline in performance. This change in performance can be attributed to the use of shorter timestamps in the OIL test data. Specifically, each item in the test demand now has 15 timestamps, whereas Nguyen (2023) utilized 17 timestamps per item. This reduction in timestamps affects the data representation and modeling, leading to the observed differences in results. In the OIL data, there are a total of 3253 items in the test data with a total demand of zero during the test periods. As a consequence, the part fill rate of these items is zero due to the inventory setting and therefore significantly impacts the average-AFR. It explains why all trade-off curves between average-AFRs and TFRs of all models are concentrated around 0.45 and 0.55. With 3253 out of 7644 items (representing approximately 42.6% of all items) having zero total demand, the average-AFR is affected substantially. As a result of this data composition, the maximum achievable average-AFR in this data set is estimated to be around 0.574.

Although the average-AFRs are low, the impact of items with 0 total demand on the total-AFR is relatively less significant. Items with higher demand have a greater influence on the total-AFR compared to items with low demand. This observation is also evident in Figure 9c, where the trade-off curves of all models between total-AFRs and TFRs are higher than the trade-off curves between average-AFRs and TFRs. Similar to Figure 9A, it is observed that the trade-off curves of ESRNN and LSTM barely change as the TFR increases. One possible explanation for this phenomenon is that certain items, depending on their forecast μ and variance σ^2 values, generate high base-stock levels even at low TFRs, while for other items, minimal stock is maintained. To illustrate this, let's consider a few items from ESRNN on TFRs of 0.80 and 0.90 for the first 10 forecasting periods: the 3137-th item with $\mu = 0.015$ and $\sigma = 0.226$, the 4603-th item with $\mu = 0.009$ and $\sigma = 0.243$, the 35-th item with $\mu = 0.010$ and $\sigma = 3.252$, the 84-th item with $\mu = 0.010$ and $\sigma = 0.714$.

Table 12: Individual Base Stock Levels in OIL - ESRNN. The TFR is 0.80, these items have atleast one positive demand in the test periods.

Item	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8	Period9	Period10
3137-th Item	1	3	4	4	4	4	4	4	4	4
4603-th Item	1	5	5	5	5	5	5	5	5	5
35-th Item	20	876	891	891	891	891	933	933	1116	1181
84-th Item	6	37	37	37	37	37	37	87	87	87

Table 13: Individual Base Stock Levels in OIL - ESRNN. The TFR is 0.90, these items have atleast one positive demand in the test periods.

Item	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8	Period9	Period10
3137-th Item	1	5	5	6	6	6	6	6	6	6
4603-th Item	1	7	7	7	7	7	7	7	7	7
35-th Item	20	1341	1345	1345	1345	1423	1423	1716	1794	1794
84-th Item	6	58	58	58	58	58	138	138	138	138

Tables 12 and 13 demonstrate that although the trade-off curves between total-AFRs and TFRs appear to be relatively flat, the base stock levels of all items increase significantly with higher TFRs. Notably, the 35-th and 84-th items exhibit extremely high base stock levels compared to the other two items. This difference is likely attributed to the relatively high observed variance σ^2 in the historical demand of these two items (the 35th item has 4 demand occurrences, with one of them being larger than 80), as their forecast μ values are comparable. It is assumed that most items have a very low forecast μ, considering the mean forecast $\mu_{mean} = 0.015$ predicted by ESRNN. Consequently, an increase in the variance of historical demands will have a substantial impact on the base stock levels. To simplify the analysis, it is considered that the base stock level experiences a significant increase when the standard deviation $\sigma > 0.5$. Approximately 3400 items among all items have a standard deviation $\sigma > 0.5$. Assuming that 42.6% of these items have no demand in the test periods, there remain 1950 items with extremely high base stock levels. Importantly, it should be noted that, as shown in Tables 12 and 13, these extremely high base stock levels exist already at lower TFRs. As the TFR increases, the base stock levels become even higher, but they are not expected to contribute further to total-AFRs, as items with high demand are already fully supplied at earlier TFRs. The observed trade-off curves' flatness in relation to total-AFRs and TFRs reinforces the idea that once the base stock levels for high-demand items are set at lower TFRs, further increasing the TFR does not significantly impact the total-AFR, as these items are already supplied.

Considering the mean demand size CV^2 of OIL is 0.18, if we assume that the mean demand size in the test set is comparable, most items will still be fully supplied despite their low observed variance. It could be possible that only a limited number of items, characterized by high demand during the test periods, but with low historical demand variability, might not be fully supplied.

This could result in the trade-off between total-AFRs and TFRs appearing relatively flat due to the limited impact of these items on the overall total-AFR. In the case of ESRNN, extremely high base stock levels are often generated for items with a low forecast, combined with a relatively high variance from the historical demand. In this context, the question arises whether the base stock levels change when the forecast further increases. For this analysis, we consider the forecast from Willemain, where the mean forecast $\mu_{mean} = 0.716$ compared to ESRNN ($\mu_{mean} = 0.015$).

Let's consider base stock levels for the same items but with a different forecast μ and the same variance σ^2 from the historical demand at a TFR of 0.80: the 3137-th item with $\mu = 0.134$ and $\sigma = 0.226$, the 4603-th item with $\mu = 0.097$ and $\sigma = 0.243$, the 35-th item with $\mu = 0.454$ and $\sigma = 3.252$, the 84-th item with $\mu = 0.408$ and $\sigma = 0.714$.

Table 14: Individual Base Stock Levels in OIL - Willemain. The TFR is 0.80, these items have atleast one positive demand in the test periods.

Item	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8	Period9	Period10
3137-th Item	1	1	1	1	1	1	1	1	1	1
4603-th Item	1	1	1	1	1	1	1	1	1	1
35-th Item	20	38	43	43	43	43	51	51	65	65
84-th Item	3	3	3	3	3	3	3	6	6	6

Based on the observations from Table 14, it is clear that an increase in the forecast leads to a reduction in the base stock level, as seen in all four items. Conversely, it appears that as the variance increases, the base stock level rises as well.

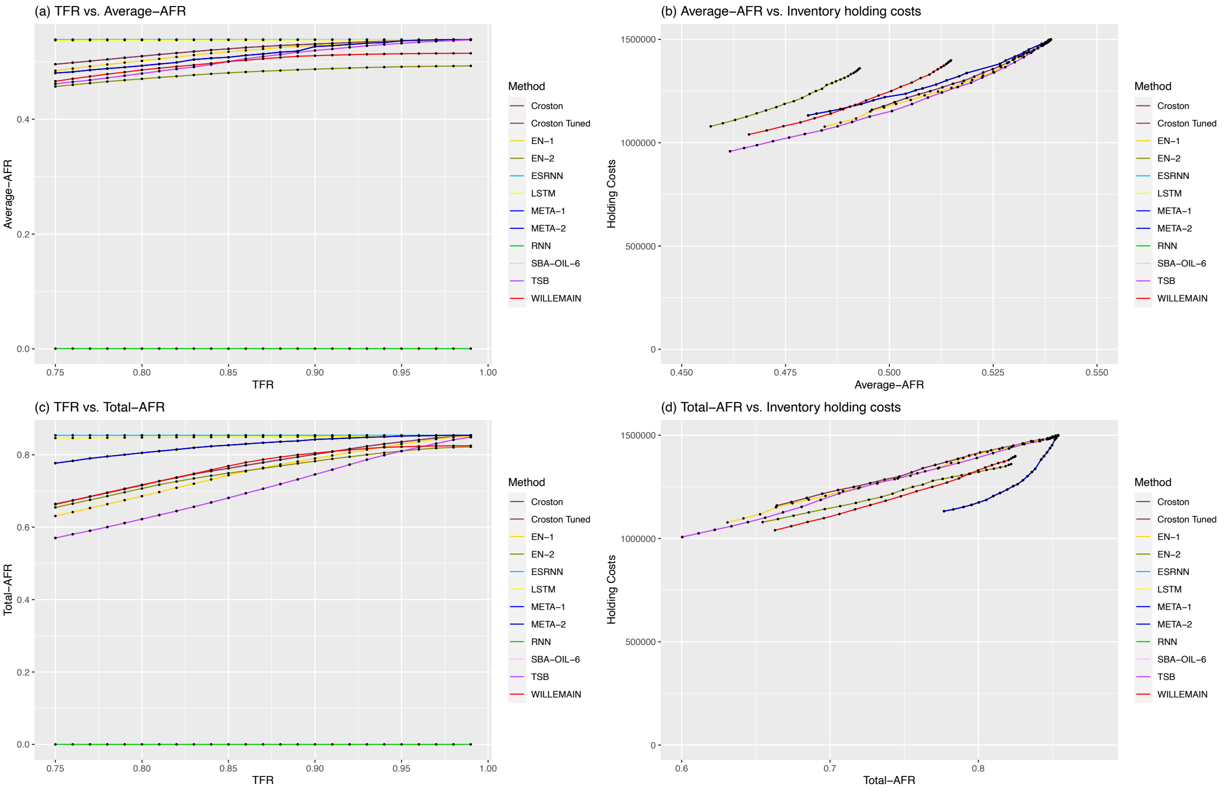

Figure 9: OIL Inventory Performance

7.2.7 MAN Inventory Performance

In the MAN data, which consists of items classified as intermittent and lumpy, with a few falling under the categories of erratic and smooth, similar patterns emerge when comparing the findings to the previous observations from the OIL data. Based on Figures 10a and 10c, ESRNN consistently achieves higher AFRs at each TFR. However, this model incurs higher inventory costs at each AFR compared to other models. LSTM emerges as the second-worst performing model in the trade-off between AFRs and TFRs, incurring higher inventory holding costs compared to other models. It appears that the Croston and its variations are superior in the trade-off between average-AFRs and inventory holding costs, while both Meta-learners outperform other models in the trade-off between total-AFRs and inventory holding costs.

Similarly to the findings in the OIL data, the trade-off between AFRs and TFRs appears to be flat for ESRNN, LSTM, and RNN in the MAN data. However, the performance in this context will also depend on whether demand is observed in the test periods and the extremely high base stock levels generated for items. There are 176 items with no demand in the test periods, meaning the maximum overall achievable average-AFR is 0.874 since the fill rate for these items is zero due to the inventory setting. As explained earlier, this will have a relatively less impact on the total-AFR as these items do not contribute significantly to the total-AFR. To illustrate this further, we will also run the base stock level for four items using the forecasts of ESRNN to determine if, similar to the observations in the OIL data, extremely high base stock levels are generated based on the forecast μ and the variance σ^2.

Let's consider a few items from ESRNN at the TFR of 0.80 for the first 10 forecasting periods: the 86-th item with $\mu = 0.010$ and $\sigma = 0.449$, the 132-th item with $\mu = 0.010$ and $\sigma = 22.090$ (please refer to Table 20 in the Appendix for the size of the historical demand for this item), the 1189-th item with $\mu = 0.518$ and $\sigma = 2.23$, the 340-th item with $\mu = 0.009$ and $\sigma = 0.358$.

Table 15: Individual Base Stock Levels in MAN - ESRNN. The TFR is 0.80, these items have atleast one positive demand in the test periods.

Item	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8	Period9	Period10
86-th Item	10	16	16	16	16	16	16	16	16	16
132-th Item	100	31746	31746	31746	31746	31746	31746	33055	33117	36120
1189-th Item	10	12	12	12	12	14	14	14	14	14
340-th Item	2	15	15	15	19	19	19	19	19	19

We also assume that, as the TFR increases, the base stock levels tend to rise accordingly. For the 132-th item, an exceptionally high base stock level is generated, likely due to the significant variability observed in the historical demand, given the very low value of the forecast. Moreover, when comparing the base stock level of the 1189th item to that of the 86th item, despite the variance being four times larger, the difference can be attributed to the higher value of forecast, as observed earlier in Table 14. The mean forecast μ_{mean} of all items of ESRNN is equal to 0.022, and the mean standard deviation σ_{mean} is 17.28. This implies that all items, including those with high demand, are fully supplied, resulting in the highest trade-off between AFRs and TFRs. It is important to know that these extremely high base stock levels are already generated at earlier TFRs, and further increasing the TFRs will not significantly contribute to the AFRs. However, there are a few items with low variability in historical demand but with high demand in test periods, for which the demand will not be fully met. It is expected that the number of such items is relatively low, and their impact on the total-AFR is negligible, leading to a relatively flat curve for ESRNN and LSTM.

The question arises again whether the base stock levels change as the forecast further increases. For this analysis, we compare the forecast from Willemain, where the overall mean forecast $\mu_{mean} = 6.286$, with ESRNN's mean forecast ($\mu_{mean} = 0.022$). Let's examine the base stock levels for the same items but with different forecasts while maintaining the same variance from the historical demand. We will use a target fill rate (TFR) of 0.80 for all items. Specifically, we consider the 86-th item with $\mu = 0.091$ and $\sigma = 0.449$, the 132-th item with $\mu = 7.430$ and $\sigma = 22.090$, the 1189-th item with $\mu = 1.491$ and $\sigma = 2.23$, and the 340-th item with $\mu = 0.325$ and $\sigma = 0.358$.

Table 16: Individual Base Stock Levels in MAN - Willemain. The TFR is 0.80, these items have atleast one positive demand in the test periods.

Item	Period1	Period2	Period3	Period4	Period5	Period6	Period7	Period8	Period9	Period10
86-th Item	2	2	2	2	3	3	3	3	3	4
132-th Item	53	53	53	53	53	53	53	53	53	53
1189-th Item	7	7	7	7	7	7	7	7	7	7
340-th Item	1	1	1	1	1	1	1	1	1	1

According to Table 16, consistent with Table 15, an increase in the forecast leads to a decrease in the base stock levels for all items. This observation is similar to what we found in the OIL data, where higher forecasts corresponded to lower base stock levels, while an increase in the variance led to higher base stock levels. This phenomenon explains why the trade-off curves between AFRs and TFRs for Willemain and other models are not flat, as the smaller base stock levels resulting from the increase in the forecast lead to varying slopes in the trade-off curves. Consequently, increasing the TFR can contribute further to items with high demand. The final observation is that RNN performs poorly in the trade-off between AFRs and TFRs. It appears that, for many items (but not necessarily all), negative forecasts result in their base stock levels being set to 0, making it impossible to fulfill any demand during the test periods for these specific items.

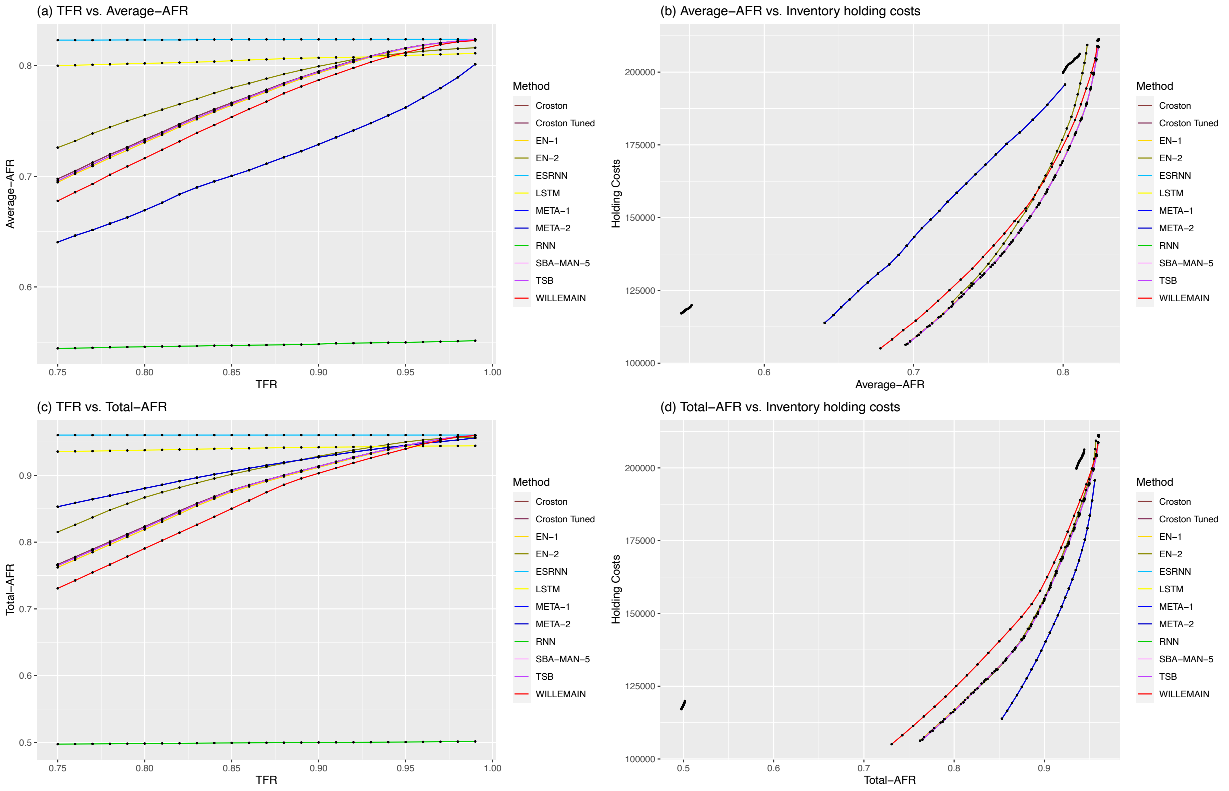

Figure 10: MAN Inventory Performance

7.2.8 BRAF Inventory Performance

Similar to the MAN data, the BRAF data primarily consists of items classified under intermittent and lumpy demand patterns. Among the methods analyzed, ESRNN, LSTM, and RNN emerge as the worst performing models in the trade-off between AFRs and TFRs, incurring higher inventory holding costs compared to other models. Willemain exhibits the worst performance in the trade-off between average-AFRs and TFRs, as well as in the trade-off between average-AFRs and inventory holding costs. Moreover, in the trade-off between total-AFRs and TFRs, Willemain is inferior to most other models. Nevertheless, in the trade-off between total-AFRs and inventory holding costs, Willemain stands as the second-best performer up to the average-AFR of 0.79. Croston and its variations demonstrate the best inventory performance in the trade-off between average-AFRs and inventory holding costs. On the other hand, Meta-learners exhibit the lowest costs in the trade-off between total-AFRs and TFRs, and they also show strong performance in the trade-off between total-AFRs and TFRs. Consistent with previous findings, ESRNN, LSTM, and RNN incur the highest inventory costs, likely due to the extremely high base stock levels they maintain.

Consistent with previous findings in the OIL and MAN data sets, the trade-off curves of ESRNN, LSTM, and RNN in Figures 11a and 11c appear to be flat. This behavior can be attributed to the generation of extremely high base stock levels when the forecast is low and the variance in historical demands is high. Let's compare the mean forecast $\mu_{mean} = 0.015$ of ESRNN with the mean forecast $\mu_{mean} = 1.588$ of Willemain, while maintaining the overall mean standard deviation σ of historical demands at 5.852. We expect that, under the ESRNN method, many more extremely high base stock levels will be generated for most items compared to Willemain, primarily due to the significant difference in the mean forecast. As a result, most low-demand and high-demand items will be fulfilled at earlier TFRs, and an increase in the TFR will not significantly contribute to the AFR since the base stock levels are already extremely high at earlier TFRs. This phenomenon leads to a flatter curve in the trade-off between AFRs and TFRs. In contrast, under Willemain, the increase in TFR is expected to contribute to a higher AFR, as the base stock levels are generally lower than those under ESRNN in this data set.

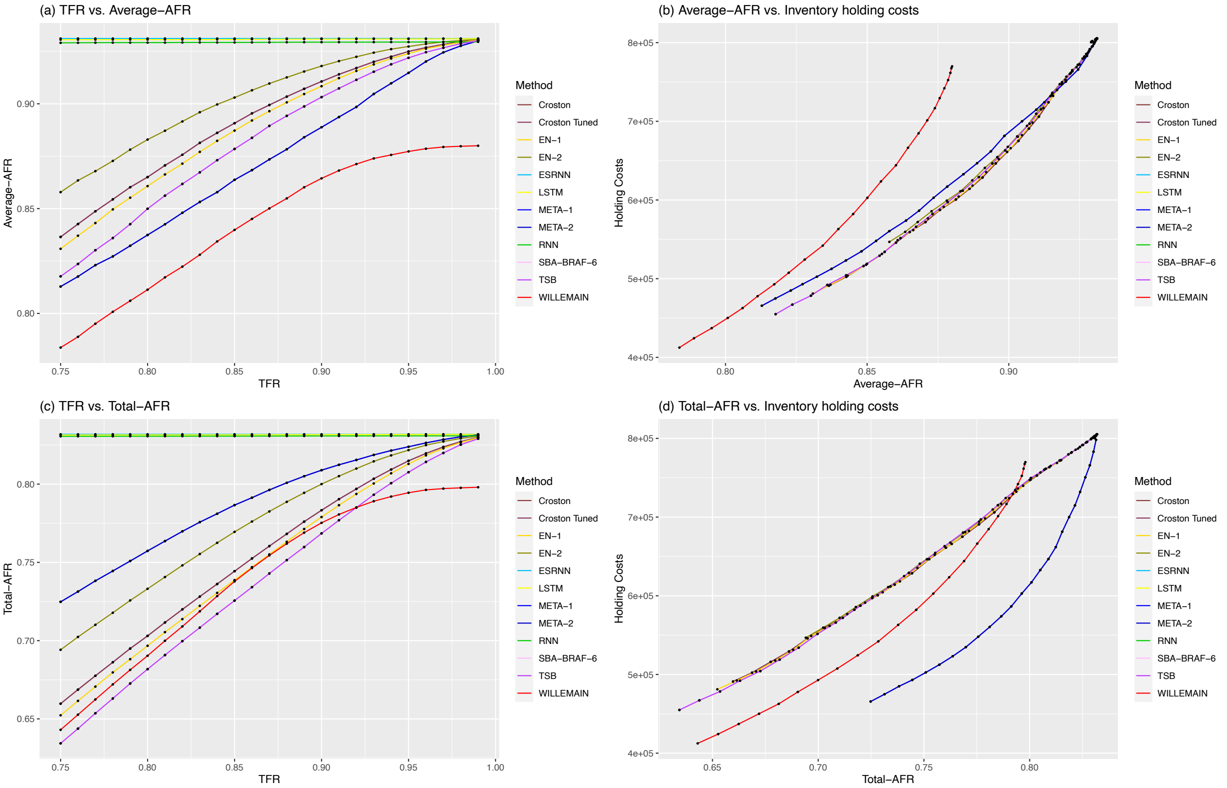

Figure 11: BRAF Inventory Performance

7.2.9 Overall Findings Inventory Performance

Finding 8. The analysis of inventory performance across all data sets reveals that ESRNN, LSTM, and RNN exhibit the worst inventory performance, incurring higher inventory holding costs compared to other models. Although these methods generally excel in the trade-off between AFRs and TFRs, this indicates that they tend to maintain excessive inventory levels, resulting in higher inventory costs when compared to other models. An exception is observed in the AUTO data, where they perform better in the trade-off between total-AFRs and inventory holding costs. Conversely, Willemain demonstrates the best inventory performance in the SIM1, SIM3, and SIM4 data sets, requiring slightly fewer inventory holding costs to achieve the same AFR compared to other models. Additionally, Willemain outperforms other models in the AUTO data, but only in the trade-off between total-AFRs and inventory holding costs. Ensemble 1 exhibits the overall best inventory performance in the SIM2 data set. In the OIL, MAN, and OIL data sets, Meta-learners show the best inventory performance, particularly in the trade-off between total-AFRs and inventory holding costs. On the other hand, Croston and its variations exhibit the best performance in the trade-off between average-AFRs and inventory holding costs. This suggests that Meta-learners and Croston based models achieve lower costs while attaining the same total-AFR (Meta-learners) or average-AFR (Croston and its variations) compared to other methods.

Finding 9. Higher accuracy does not guarantee better inventory performance. This pattern becomes evident in the simulated data sets where Meta-learners demonstrate superior forecasting performance, but are outperformed by Willemain in inventory control assessment. Similarly, in the OIL data, Willemain exhibits the best forecasting performance, but it is outperformed by Croston and its variations in the trade-off between average-AFRs and inventory holding costs. Conversely, the worst-performing models in terms of accuracy, namely ESRNN, LSTM, and RNN, generally exhibit the worst inventory performance.

Finding 10. The implementation of inventory control is highly sensitive to small adjustments, as observed in the OIL data, where Nguyen (2023) used two more timestamps in the test periods. This change resulted in a increased likelihood of positive demand in the test periods, leading to less items being assigned a fill rate of 0. Therefore, by having fewer timestamps in the test periods, this adjustment significantly impacted the average-AFR. In such cases, a question arises as to whether we should assign a fill rate of 1 to items that have no demand in the test periods. This decision could have a substantial impact on inventory performance, particularly in the OIL and MAN data sets.

Finding 11. In the simulated data, the difference between evaluating inventory control based on average-AFR and total-AFR is minimal, as both measures are approximately equal at the given TFR, resulting in similar holding costs. While ESRNN is slightly more expensive than other models, the difference in holding costs is not significant compared to the industrial data sets. However, in industrial data sets, the choice of metric has a significant impact on the inventory holding costs. The discrepancy between the two metrics could be attributed to models like Meta-learners, which are better at predicting cheaper items with high demand, compared to Croston and its variations, which excel at forecasting expensive items with low demand. These findings suggest that not only the intermittency between industrial and simulated data sets does not match, but also the demand size does not align. It is recommended to introduce more variation in the demand when generating simulated data in future research. Moreover, it has been found that the time period of five years in the industrial data is often too short to observe the demand adequately. A longer time period would provide more demand data, leading to more stable demand patterns.

Finding 12. As mentioned earlier, the base stock level $R_{i,r,k}$ is zero when the forecast $\mu \leq 0$. This implies that the RNN model is not suitable for the MAN and OIL data sets, since forecast μ values of most items are below zero. Specifically, in the case of the OIL data, nearly all forecasts predicted by the RNN model are below zero, resulting in setting the base stock level to zero for all items. Similarly, in the MAN data, although there are a few items with positive forecasts, the overall mean forecast for the entire data set is negative.

Finding 13. Machine learning methods like ESRNN, LSTM, and RNN demonstrate flat trade-off curves between AFRs and TFRs in the OIL, MAN, and BRAF data sets. This behavior can be attributed to the generation of extremely high base stock levels by these methods. As previously explained, the base stock levels for forecasted demands by ESRNN, LSTM, and RNN are already set high at lower TFRs. Consequently, an increase in the TFR leads to even higher base stock levels, but it does not significantly impact the AFR since most demands are already fulfilled at earlier TFRs. This dynamic is likely the reason why the trade-off curves appear flat in these cases.

Finding 14. Under the assumption that the forecast follows the gamma distribution and using the inventory setting as described in this paper, a decrease in the forecast leads to a rise in the base stock level, while an increase in the forecast causes it to decline. Similarly, as the variance in the historical demand increases, it results in higher base stock levels, and the base stock level decreases when the variance becomes smaller. The combined effect of these two parameters can either amplify or diminish the impact on the base stock level, for example, when the forecast decreases and the variance increases, or when the forecast increases and the variance increases. To illustrate this, let's consider three items given a TFR of 0.85:

(1) Item 1 ($\mu = 0.01$ and $\sigma = 2$), this results in $k = 0.0025$ and $\alpha = 0.000025$, the loss target $(1 - TFR) \cdot \mu = (1 - 0.85) \cdot 0.01 = 0.0015$, the base stock level should be set to at least 398, this results in $ESC = 0.001496 <$ loss target.

(2) Item 2 ($\mu = 1$ and $\sigma = 2$), this results in $k = 0.25$ and $\alpha = 0.25$, the loss target $(1 - TFR) \cdot \mu =$

$(1-0.85) \cdot 1 = 0.15$. To equate again the loss function to the ESC, a base stock level of at least 5 is required.

(3) Item3 ($\mu = 1$ and $\sigma = 6$), this results in $k = 0.028$ and $\alpha = 0.028$, the loss target is still 0.15, a base stock level of 37 is needed to equate the loss function to the ESC.

These examples demonstrate that a decrease in the forecast or an increase in the variance would likely imply a higher base stock level to meet the loss target. This observation aligns with the findings of ESRNN and Willemain in the cases of OIL, MAN, and BRAF data sets. It is logical that the loss target becomes much lower when the forecast is extremely small, leading to the expectation of requiring a higher base stock level to meet it. However, the examples also demonstrate that even with the same forecast μ, but a higher variability in historical demands, a higher base stock level is generated while the loss target remains unchanged. This suggests that the reduction in the shape and rate parameters from the gamma distribution due to a higher variance (or a smaller forecast) results in a smaller decrease in the ESC per unit increase in the base stock level. As a consequence, higher base stock levels are expected to be needed to meet the loss target.

Lastly, High base stock levels could be mitigated by exploring alternative probability distributions, such as the normal or negative binomial distribution. It would be interesting to evaluate these distribution functions and determine if they lead to a improved inventory performance, especially when the forecast is (extremely) low.

7.3 Comparative Studies

As mentioned in section 2.4, Pinçe et al. (2021) found that SBA exhibited better overall accuracy performance than Croston in their comparison of 53 spare parts forecasting studies. De Haan (2021) extended the comparative studies by benchmarking Croston, TSB, SBA, Willemain, MLP, and LightGBM. These methods were implemented on the same empirical data setting as this paper, including four simulated data sets with unique demand patterns: erratic, lumpy, smooth, and intermittent, as well as four industrial data sets. De Haan (2021) found that no single model outperformed others in both accuracy and inventory measures. Specifically, SBA demonstrated the best accuracy performance overall, while it lagged behind Willemain in terms of inventory performance. However, in cases where demand exhibits extreme intermittency, such as in the MAN and OIL data, MLP and LightGBM performed better in terms of inventory performance compared to Willemain.

Furthermore, De Haan (2021) found that the performance of a model depended on the measure being evaluated and the type of data. No model proved to be superior across all types of data sets, accuracy measures, and inventory performance. In line with the findings of Nguyen (2023), the study observed that SBA demonstrated the best overall accuracy performance, while LightGBM performed the worst. In terms of inventory control performance, Willemain was found to be the best, except for data sets with high intermittency such as MAN, OIL, and BRAF. For these specific data sets, MLP and LightGBM methods were found to have the best performance based on the applied inventory control measures.

Similar to previous studies (De Haan, 2021; Pinçe et al., 2021; Nguyen, 2023), the findings of this research indicate that no single model consistently achieves the highest forecasting performance across all accuracy measures and types of data sets. Through cumulative voting, Meta-learner 1 receives the highest number of votes and demonstrated superior performance in SIM1, SIM2, SIM3, SIM4, and BRAF data sets, as evaluated on MSE, RMSE, and RMSSE.

In contrast to the findings of previous studies by De Haan (2021) and Nguyen (2023), SBA is not found to be superior in the forecasting performance when the data input is non-aggregated. However, the aggregation of the industrial data sets, except for the AUTO data, led to improved accuracy performance for SBA-OIL-6, SBA-MAN-5, and SBA-BRAF-6 models. These models even ranked first in the OIL and MAN data sets based on MSE, RMSE, and RMSSE metrics. Furthermore, the results presented in this paper do not support the claim that machine learning methods, such as ESRNN, LSTM, and RNN are superior when dealing with data sets characterized by high intermittency as observed in the OIL, MAN, and BRAF data sets.

Lastly, the results demonstrate that Willemain exhibits a slightly better inventory performance in simulated data sets and is superior in the AUTO data. Meta-learners prove to be superior in the trade-off between total-AFR and inventory holding costs in the OIL, MAN, and BRAF data sets, while Croston and its variations are found to be superior in the trade-off between average-AFR and inventory holding costs in the same data sets.

8 Conclusion

Based on the literature review (Croston, 1972; Syntetos, Boylan, 2005; Willemain et al., 2004; Teunter et al., 2011; Kourentzes, 2018), the M4 and M5 forecasting competitions (Makridakis, Spiliotis, & Assimakopoulos, 2022), as well as the comparison studies (De Haan, 2021; Pinçe et al., 2021; Nguyen, 2023), various methods have been utilized in spare parts forecasting. These methods have been assessed for their accuracy and inventory performance across different types of data sets. In this research, the following methods are implemented: Croston, Croston optimized, TSB, SBA, RNN, LSTM, ESRNN, Willemain, equal-weighted combination (ensemble 1 and 2), and stacking through XG-Boost (Meta-learners 1 and 2). These methods are applied to simulated data sets with erratic, lumpy, smooth, and intermittent demand patterns, as well as real industrial data sets including BRAF, MAN, AUTO, and OIL.

The sub-research question addressed in this study was:

Does aggregating observations into larger intervals improve the accuracy of the forecasting model?

Based on the findings presented in Table 11, the accuracy performance of SBA is increased across all accuracy measures in the OIL, BRAF, and MAN data. However, no improvements are observed in the AUTO and the simulated data. These findings are consistent with the research conducted by Nikolopoulos (2011), which suggests that aggregating time series can enhance forecast accuracy when the data is characterized by a high mean inter-demand interval. In this study, ADIDA is found to be effective in improving the forecasting measure, particularly when applied to data sets with extremely high mean inter-demand intervals. In addition, despite the higher intermittency in the MAN data, the reduction in forecasting error is more significant in the OIL data, which has lower intermittency. This difference is likely due to the specific demand patterns, occurrences, and timing of positive demand observed in the OIL data.

The main research question addressed in this study was:

Do ensemble models outperform Croston and its variations in terms of forecasting accuracy and inventory performance on 4 simulated and 4 industrial spare parts demand data?

The results show, no single model consistently demonstrated superiority across all forecasting measures and data sets. However, cumulative voting results in Table 10 reveal that Meta-learner 1 emerges as the best-performing model in terms of MSE, RMSE, and RMSSE. When compared to Croston and its variations, Meta-learner 1 outperforms them in the SIM1, SIM2, SIM3, SIM4, OIL, and BRAF data sets. Regarding the inventory performance, similar to the evaluation of accuracy performance, there is no single model that consistently outperforms all other models on every data set. Willemain demonstrates a slightly better inventory performance compared to other models in simulated data sets, except for SIM2 data, and is also superior in the trade-off between average-AFRs and inventory holding costs in the AUTO data. Meta-learners exhibit superiority only in the trade-off between total-AFRs and inventory holding costs in OIL, MAN, and BRAF data sets, while Croston and its variations incur the lowest costs to achieve the same average-AFR. On the other hand, ESRNN, LSTM, and RNN show the worst inventory performance in all data sets, primarily due to their tendency to hold excessive inventory, leading to significantly higher inventory holding costs compared to other forecasting models. The findings of this study partially align with previous research (De Haan, 2021; Nguyen, 2023), indicating that Willemain shows superior inventory performance as long as the data is not characterized by high intermittency. However, it is important to note that in previous studies, machine learning methods like LightGBM and MLP were found to be superior in inventory control assessment for data sets with high intermittency. In contrast, the machine learning methods used in this paper, namely ESRNN, LSTM, and RNN, exhibit the worst inventory performance in data sets such as OIL, MAN, and BRAF.

9 Limitations

In future research, it is important to continue examining the effectiveness of spare parts forecasting methods using a combination of empirical and simulated data. While simulated data allows for controlled experiments and the assessment of different demand patterns, it is crucial to validate the findings by comparing them with real-world industrial data. This will provide a clearer understanding of how well the methods perform in practical scenarios. One important aspect to consider is the generation process of simulated data. In cases where extreme values are observed in industrial data, such as in MAN, BRAF, and OIL data, it would be better to generate simulated data that closely resembles these characteristics, which indicates longer zero-demand periods and higher demand sizes to accurately simulate the characteristics in real-world scenarios.

Additionally, it is important to address the issue of zero-demand occurrences in certain test data, such as OIL and MAN. These zero-demand occurrences can have a significant impact on forecasting accuracy and inventory performance. One potential approach to mitigate this is by either adding demand occurrences or removing the item with no demands in the test data.

Moreover, it would be valuable to conduct a more comprehensive analysis of the demand pattern, including the number of demand occurrences, the timing of demand, and whether there are subsequent positive demands following an initial positive demand. This more detailed classification of demand patterns can provide more insights beyond the general categories of erratic, lumpy, smooth, or intermittent. By examining these specific aspects by different methods, a more thorough understanding of demand behavior can be obtained.

Regarding the implemented methods, it is recommended to explore a broader range of combined forecasting methods. In this study, only four combined methods are considered, but other combinations can further improve accuracy and inventory performance in spare parts forecasting. This can involve expanding the pool of methods, exploring different optimization techniques beyond XGBoost, or even assigning weights to each model using cross-validation. Also, it is worth knowing that in this research, only SBA is used to predict the aggregated simulated and industrial data sets, it is recommended to explore other methods beyond SBA to improve the forecasting accuracy when dealing with high intermittency.

Furthermore, in this research, ESRNN, LSTM, RNN are implemented and demonstrate a poor forecasting accuracy. A poor forecasting performance can be attributed to the fact that ESRNN, being a partially and globally trained model, overlooks the individual behaviors of each spare part, and a bias is introduced due to a technical limitation. The recommendation is to develop or utilize a hybrid framework specifically designed for spare parts forecasting. This framework should take into account the unique characteristics and demand patterns of spare parts, such as intermittency and individual behavior.

Extremely high base stock levels are observed in the OIL, MAN, and BRAF data sets when using the methods ESRNN, LSTM, and RNN, which are not the case with other models such as Willemain, Meta-learners, and Croston and its variations. This disparity can be attributed to the extremely low mean forecast (μ_{mean}) predicted by ESRNN, LSTM, and RNN in comparison to other models. Consequently, it results in lower shape and rate parameters for the gamma distribution, implying the need of a higher base stock level to achieve the loss target. Moreover, the study has revealed that an increase in the variance of historical demand leads to higher base stock levels. Both the decrease in the forecast and the increase in the variance contribute to lower shape and rate parameters of the gamma distribution, leading to a smaller reduction in the ESC per unit increase in the base stock level. An interesting question arises regarding the comparative impact of changes in the forecast versus changes in the variance on base stock levels. Moreover, it would be interesting to determine specific combinations of the forecast μ and the variance σ^2 at which base stock levels experience a significant increase.

Lastly, to improve the evaluation of inventory performance, future studies should consider exploring the use of probability distributions beyond the gamma distribution, avoiding of excessively high base stock levels, and consider using real lead-time information.

10 References

Babai, M. Z., Dallery, Y., Boubaker, S., & Kalai, R. (2019). A new method to forecast intermittent demand in the presence of inventory obsolescence. International Journal of Production Economics, 209, 30-41.

Bookbinder, J. H., & Lordahl, A. E. (1989). Estimation of inventory re-order levels using the bootstrap statistical procedure. IIE transactions, 21(4), 302-312.

Burgin, T.A., 1975. The gamma distribution and inventory control. Journal of the Operational Research Society 26, 507–525. URL: https://doi.org/10.1057/jors.1975.110, doi:10.1057/jors.1975.110, 740 arXiv:https://doi.org/10.1057/jors.1975.110.

Chan, F., & Pauwels, L. L. (2018). Some theoretical results on forecast combinations. International Journal of Forecasting, 34(1), 64-74.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. Operational Research Quarterly, 23(3), 289-303.

De Haan, D. (2021). Benchmarking spare part demand forecasting methods.

Durlinger, B., & van Houtum, G. J. (2017). Spare parts logistics: Trends and research challenges. Production and Operations Management, 26(2), 225-242.

Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2), 179-211.

Gutierrez, R. S., Solis, A. O., & Mukhopadhyay, S. (2008). Lumpy demand forecasting using neural networks. International Journal of Production Economics, 111(2), 409-420.

Guo, F., Diao, J., Zhao, Q., Wang, D., & Sun, Q. (2017). A double-level combination approach for demand forecasting of repairable airplane spare parts based on turnover data. Computers & Industrial Engineering, 110, 92-108.

Hasni, M., Aguir, M., Babai, M., & Jemai, Z. (2019). Spare parts demand forecasting: A review on bootstrapping methods. International Journal of Production Research, 57(15-16), 4791-4804.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

Hyndman, R. J., & Khandakar, Y. (2008). Automatic time series forecasting: The forecast package for R. Journal of Statistical Software, 27, 1-22.

Hyndman, R. J., Athanasopoulos, G., Bergmeir, C., Caceres, G., Chhay, L., O'Hara-Wild, M., Petropoulos, F., Razbash, S., Wang, E., & Yasmeen, F. (2018). Forecast: Forecasting functions for time series and linear models. R package version 8.3.

Jaganathan, S ., Srihari, & Prakash, P. K. S. (2020). A combination-based forecasting method for the M4-competition. International Journal of Forecasting, 36(1), 98-104.

Kolassa, S. (2011). Combining exponential smoothing forecasts using Akaike weights. International Journal of Forecasting, 27(2), 238-251.

Kostenko, A. V., & Hyndman, R. J. (2006). A note on the categorization of demand patterns.

Journal of the Operational Research Society, 57(10), 1256-1257.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). M5 accuracy competition: Results, findings, and conclusions. International Journal of Forecasting, 38(4), 1346-1364.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., & Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. International Journal of Forecasting, 36(1), 86-92.

Moors, J. J. A., & Strijbosch, L. W. (2002). Exact fill rates for (R, s, S) inventory control with gamma distributed demand. Journal of the Operational Research Society, 53(11), 1268-1274.

Mukhopadhyay, S., Solis, A. O., & Gutierrez, R. S. (2012). The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. Journal of Forecasting, 31(8), 721-735.

Nguyen, K., De Haan, D., (2023). A comparison of several spare parts demand forecasting methods.

Pinçe, Ç., Turrini, L., & Meissner, J. (2021). Intermittent demand forecasting for spare parts: A critical review. Omega, 102513.

Porras, E., & Dekker, R. (2008). An inventory control system for spare parts at a refinery: An empirical comparison of different re-order point methods. European Journal of Operational Research, 184(1), 101-132.

Sak, H., Senior, A., & Beaufays, F. (2014). Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128.

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. International Journal of Forecasting, 21(2), 303-314.

Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research, 214(3), 606-615.

Van der Laan, E. A., Dekker, R., & Salomon, M. (2014). Spare parts inventory control: A literature review. European Journal of Operational Research, 237(2), 408-420.

Van Wingerden, E., Basten, R. J. I., Dekker, R., & Rustenburg, W. (2014). More grip on inventory control through improved forecasting: A comparative study at three companies. International Journal of Production Economics, 157, 220-237.

Willemain, T. R., Smart, C. N., Shockor, J. H., & DeSautels, P. A. (1994). Forecasting intermittent demand in manufacturing: a comparative evaluation of Croston's method. International Journal of Forecasting, 10(4), 529-538.

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. International Journal of Forecasting, 20(3), 375-387.

Zhou, C., & Viswanathan, S. (2011). Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. International Journal of Production Economics, 133(1), 481-485.

Zhu, S., Dekker, R., Van Jaarsveld, W., Renjie, R. W., & Koning, A. J. (2017). An improved method for forecasting spare parts demand using extreme value theory. European Journal of Operational Research, 261(1), 169-181.

11 Githubs

De Haan, D.(2021). A respository with both industrial and simulated datasets on spare part demand forecasting, with the intention of benchmarking them. Retrieved from https://github.com/danieldehaan96/spdf

Nixtla (2023). Scalable and user friendly neural forecasting algorithms. Retrieved from https://github.com/Nixtla/neuralforecast

Nixtla (2023). Lightning fast forecasting with statistical and econometric models. Retrieved from https://github.com/Nixtla/statsforecast

Nguyen, K.(2023). Forecasting the demands of spare parts, using 7 different methods on 8 data sets (industrial and simulated). Retrieved from https://github.com/KhueNguyenTK/Spare-Part-Demand-Forecasting

Pmontman (2020). R package for Feature-based Forecast Model Averaging. Retrieved from https://github.com/pmontman/fforma

12 Appendix

Table 17: Descriptive statistics Industrial Data Sets

Data set	MinSale	MeanSale	MaxSale	StDevSale	MinPrice	MeanPrice	MaxPrice	StDevPrice
MAN	0.007	24.224	4599.653	139.294	0.085	19.958	297.537	31.356
BRAF	0.036	1.442	65.083	3.617	0.001	102.321	9131.992	373.334
AUTO	0.542	4.450	129.167	7.573	32.596	946.176	7772.856	1369.320
OIL	0.036	0.629	232.727	4.016	0.010	355.848	20493.170	1076.121

Table 18: Descriptive Statistics Simulated Data Sets

Data set	CV2	p	MeanSale	StDevSale	MeanPrice	StDevPrice
SIM1	0.75	1.00	10.014	1.125	2129.298	246.061
SIM2	0.80	1.50	6.662	1.124	1416.519	254.797
SIM3	0.30	1.05	9.496	0.737	2019.207	159.058
SIM4	0.25	1.45	6.897	0.812	1466.476	180.186

Table 19: Forecasting Performance

Model	Metric	AUTO	OIL	MAN	BRAF	SIM1	SIM2	SIM3	SIM4
Croston	MSE	86.107	170.642	12982.277	205.379	79.468	78.686	35.181	40.393
	RMSE	9.279	13.063	113.940	14.331	8.914	8.871	5.931	6.356
	MASE	0.615	1.041	0.880	0.929	0.731	0.759	0.719	0.759
	RMSSE	1.703	10.923	11.439	5.020	1.008	1.049	0.940	0.921
Croston_T	MSE	85.724	168.487	12982.191	205.357	79.472	78.698	35.200	40.432
	RMSE	9.259	12.980	113.939	14.330	8.915	8.871	5.933	6.359
	MASE	0.611	1.027	0.880	0.929	0.731	0.759	0.719	0.760
	RMSSE	1.699	10.853	11.439	5.019	1.008	1.049	0.940	0.922
TSB	MSE	88.329	241.594	12977.837	208.891	79.281	78.547	35.071	40.258
	RMSE	9.398	15.543	113.920	14.453	8.904	8.863	5.922	6.345
	MASE	0.627	1.385	0.883	0.995	0.731	0.761	0.718	0.759
	RMSSE	1.725	12.996	11.437	5.062	1.007	1.048	0.938	0.920
SBA	MSE	85.607	170.916	12958.075	205.273	79.538	78.757	35.295	40.463
	RMSE	9.252	13.073	113.834	14.327	8.918	8.875	5.941	6.361
	MASE	0.608	1.029	0.891	0.922	0.731	0.757	0.719	0.760
	RMSSE	1.698	10.931	11.428	5.018	1.009	1.050	0.941	0.922
RNN	MSE	137.574	163.775	14275.197	202.786	83.357	88.743	35.096	41.385
	RMSE	11.729	12.797	119.479	14.240	9.130	9.420	5.924	6.433
	MASE	0.722	1.267	0.749	0.435	0.712	0.724	0.714	0.766
	RMSSE	2.153	10.700	11.995	4.988	1.033	1.114	0.939	0.933
LSTM	MSE	137.001	159.803	14258.628	202.778	83.642	87.792	35.140	41.806
	RMSE	11.705	12.641	119.409	14.240	9.146	9.370	5.928	6.466
	MASE	0.725	1.085	0.824	0.433	0.712	0.726	0.715	0.768
	RMSSE	2.148	10.570	11.988	4.988	1.035	1.108	0.939	0.937
ESRNN	MSE	150.242	152.068	14198.584	202.757	98.917	123.762	88.268	101.197
	RMSE	12.257	12.332	119.158	14.239	9.946	11.125	9.395	10.060
	MASE	0.834	0.456	0.649	0.433	0.807	0.871	0.774	0.792
	RMSSE	2.250	10.311	11.963	4.988	1.125	1.258	1.063	1.138
Willemain	MSE	84.740	139.530	13063.914	199.971	78.241	78.718	34.966	40.769
	RMSE	9.205	11.812	114.297	14.141	8.845	8.872	5.913	6.385
	MASE	0.659	0.928	0.927	0.890	0.774	0.797	0.746	0.769
	RMSSE	1.690	9.877	11.475	4.953	1.001	1.049	0.937	0.926
Ensemble_1	MSE	86.317	182.603	12974.434	206.134	79.427	78.660	35.176	40.376
	RMSE	9.291	13.513	113.905	14.357	8.912	8.869	5.931	6.354
	MASE	0.615	1.120	0.884	0.944	0.731	0.759	0.719	0.759
	RMSSE	1.705	11.299	11.435	5.029	1.008	1.049	0.940	0.921
Ensemble_2	MSE	85.881	167.468	13087.264	201.306	78.971	78.778	34.760	39.984
	RMSE	9.267	12.941	114.400	14.188	8.887	8.876	5.896	6.323
	MASE	0.597	1.100	0.832	0.840	0.721	0.742	0.714	0.756
	RMSSE	1.701	10.820	11.485	4.970	1.005	1.050	0.934	0.917
Meta-learner_1	MSE	90.864	142.519	13931.038	196.948	75.400	75.936	33.140	38.829
	RMSE	9.532	11.938	118.030	14.034	8.683	8.714	5.757	6.231
	MASE	0.633	0.849	0.935	0.778	0.734	0.765	0.710	0.748
	RMSSE	1.750	9.982	11.849	4.916	0.982	1.031	0.912	0.903
Meta-learner_2	MSE	90.864	142.519	13931.038	196.948	75.394	75.939	33.136	38.838
	RMSE	9.532	11.938	118.030	14.034	8.683	8.714	5.756	6.232
	MASE	0.633	0.849	0.935	0.778	0.734	0.765	0.710	0.748
	RMSSE	1.750	9.982	11.849	4.916	0.982	1.031	0.912	0.904

Table 20: Positive demands of the 132nd item. Obtained from the training set of the MAN data.

Period	Demand
Period4	2
Period10	20
Period20	2
Period24	12
Period32	30
Period39	10
Period40	10
Period42	10
Period45	22
Period47	42
Period51	20
Period56	20
Period65	10
Period68	50
Period70	20
Period73	100
Period87	130
Period89	20
Period93	20
Period95	100
Period99	10