Master Thesis Data Science & Marketing Analytics

Comparing spare parts demand forecasting methods

Author & Student ID number:

Yllor Hasani (640757yh)

Supervisor:

Prof. dr. ir. Rommert Dekker

Second assessor:

Prof. dr. Martijn G. de Jong

Final version:

October 30, 2023

Abstract

Comparing spare parts demand forecasting methods is an important part of the spare parts demand forecasting field. Even more so, when newer methods are introduced. In this paper, new methods are compared to older, widely-used methods. The methods compared in this paper are Croston's method, Syntetos-Boylan approximation (SBA), DLP, LightGBM, Long-short term memory (LSTM), Multi-Layer-Perceptron (MLP), Random Forest (RF), Willemain's method and Quantile regression. Every method is applied to eight different data sets. The data sets are grouped into simulated data sets or industrial data sets. The performance of the methods is measured through forecasting accuracy measures and inventory performance measures. In terms of forecasting accuracy Quantile regression was superior overall followed by MLP. Willemain's method was the overall best in terms of inventory performance. However, for lumpy demand, LSTM outperforms Willemain in terms of inventory performance. For erratic demand MLP outperforms Willemain. Whereas MLP was the second-best performer in terms of forecasting accuracy, LSTM did not stand out in terms of forecasting accuracy. We then compared the results to the reviewed recent literature and found them to be comparable. Through this research, several findings stand out, the performance measure used and the data set category have an influence on the results. Data cleaning plays a crucial role and that hyper parameter tuning takes time and requires prior knowledge.

Contents

1 Introduction

Spare parts producing companies have a responsibility of being able to replace obsolete, defective parts with new ones when needed. Some spare parts can take quite some time to be manufactured. Furthermore, when the demand for such a spare part is spontaneous and of important quantity, OEMs [1] might be caught off-guard and the spare parts production takes time to get going or if a higher amount is needed, to scale up. The waiting time for a spare part can be costly as it can cause downtime of a production or a service (Haan, 2021). The obvious solution to face the disruption of production caused by a lack of spare parts would be to have spare parts at all times and in all places in stock. However, keeping inventory is also costly. Especially, if the spare parts in the inventory are expensive and take away a considerable amount of space. Silver (1981), as described in Willemain et al. (2004), states that the demand for spare parts can be intermittent and variate between no demand at all for multiple periods to a very high demand. In other words, the demand can be infrequent but also the demand quantity can highly vary. For this reason, Willemain et al. (2004) and Syntetos et al. (2015) explain that demand forecasting is not only difficult but also highly important so that the inventory can be managed correctly. Durlinger and Paul (2015) and Callioni et al. (2005) found that in general, companies' yearly inventory holding cost can make up between 5 to 45 per cent of the costs of the inventory.

Knowing the spare parts demand in advance is not only beneficial for the company's stock management but also for its finances. This is why spare parts forecasting is such an important topic regarding a firm's sales. It determines how much you need to order. Supplying spare parts can be a vital sales advantage as the spare parts business is of high importance with high margins. Suomala et al. (2002) elaborate that the spare parts business is economically significant in many industries and can often even be considered the most profitable function of a corporation. Some industries consider product sales as a positioning opportunity so that the customers depend on the services and pull-through sales of the product company. For example, *Epson* and *Hewlett-Packard* mainly profit from the sale of toner cartridges and not from the initial printer sale, (Dennis and Kambil, 2003).

There are numerous methods for forecasting in general. However, the methods for spare parts forecasting should be able to account for high intermittency and high variability. Furthermore, as the machine learning domain is rising, ML [2] methods for spare parts demand forecasting have been developed. As for statistical methods, some ML spare parts forecasting methods perform better than others depending on the situation (Haan, 2021). This research is important

[1] Original equipment manufacturers

[2] Machine learning

as Pinçe et al. (2021) show that in the last five years, there were nearly no comparative studies papers produced related to the spare parts forecasting field. Furthermore, the ML domain is continuously advancing. So, what can new comparative papers add to the field of spare parts forecasting?

In this comparative paper, different methods of spare parts forecasting will be compared to each other with the goal of being able to deduce which method performs best for which type of demand. To measure the accuracy of demand forecasting and inventory performance, different measures for both will be used. This will allow us to also find out if different measures provide different results. The data sets used for the comparison are four industrial data sets and four simulated data sets, representing the demand, which first needs to be classified in one of the following demand classes: *Erratic, Lumpy, Smooth and Intermittent.* More details about the data sets and the classification process will be given in Section 4 Data. This leads us to the main research question:

"Which methods perform best on what kind of demand respectively for which data set?".

To answer this question, two other sub-questions come up: *"Is the performance of certain methods due to the measure used?"* and *"Do ML methods perform better in general than statistical methods?".*

In this empirical research, we first review existing spare parts demand forecasting literature in Section 2. We then present Section 3, where the research design of the paper, the methods used for the forecasting and the accuracy measures for the forecasts and inventory control performances, are presented. The methods are applied to eight different data sets, which are presented in Section 4. After applying the different methods to the data sets, the numerical results are interpreted and analysed in Section 5. This section answers the research question and its sub-questions. Last but not least, Section 6 concludes and discusses the findings and future research opportunities regarding our paper.

2 Literature Review

The structure of the literature review is inspired by Pinçe et al. (2021)'s review. In their work, they explain that the spare parts demand forecasting literature consists of three major groups with several subcategories. The three major categories are Time-series methods, contextual methods and comparative studies like this paper.

2.1 Time-series forecasting methods

The first literature category contains time-series forecasting method papers. Time-series forecasting method papers contain detailed explanations about time-series methods. Haan (2021) explains that time-series methods are built on historical data, from which they try to provide a forecast of future data. Pinçe et al. (2021) subgroups the time-series literature into three sub-categories: parametric, non-parametric and forecast improvement strategies. The latter is divided into two branches, demand classification and demand aggregation. The four demand classifications are *erratic, lumpy, smooth and intermittent*. The categorization happens based on multiple demand characteristics, which are explained in Section 4.1 Industrial data sets. In this paper, the demand is analyzed and the best performing forecasting method is recommended. Regarding data aggregation methods, they have for goal to reduce the variability of the demand (Pinçe et al., 2021).

Syntetos et al. (2012) explain that a parametric approach assumes that the lead-time demand follows a certain known distribution. Whereas non-parametric approaches, explained by Pinçe, Turrini, and Meissner, observe their lead-time demand distribution from the data. Both categories can be further sub-categorised. The parametric branch can be subdivided into Croston's modification, whether demand obsolescence is incorporated, Bootstrapping, if statistical bootstrapping is used for the parametric category. The non-parametric category is divided into three sub-categories: Bootstrapping, Neural Network and Empirical method. Those different categorizations of time-series methods are relevant for this research as they will also be used in this paper. Furthermore, the grouping allows to give a general conclusion for each category and their performance when compared.

It is noteworthy, that many forecasting methods exist and more methods are being developed, as it is a challenging scientific topic. In the next subsections, the existing literature on the methods used in this paper is reviewed. Some existing methods are briefly mentioned or not mentioned at all in our literature review, as this would be too extensive for a master's thesis.

2.1.1 Parametric approaches

The first parametric spare parts forecasting method is Croston's method. Croston's method was developed to face the inaccuracies of traditional forecasting methods such as Simple Exponential Smoothing (SES), which were caused by the predictions of periods of no demand or very low demand (Croston, 1972). Croston (1972) solves this, by splitting the demand estimate into the demand size part and inter-demand interval part. The two parts are predicted individually with SES. Being the first spare parts forecasting method, Croston is used as a benchmark for the performance comparison with other methods.

Later, Syntetos and Boylan (2005) introduced a new method named Syntetos-Boylan approximation (SBA), which is also based on SES as Croston's method. As mentioned in Pinçe et al. (2021), Syntetos and Boylan (2001) explain that Croston's method is biased. SBA corrects for the bias. A formula of SBA and an explanation of how it is different from Croston's method can be found in Section 3.1, where the methods used in this paper are elaborated. Pinçe et al. (2021) concludes that in terms of accuracy measures, SBA outperforms Croston's method for industrial spare parts data sets. After SBA, other parametric approaches, like Teunter-Syntetos-Babai (TSB) have been introduced. However, they are not used in this comparative study. The reason for this is given in Section 3.1.

Last but not least, a more recent parametric approach named DLP was introduced by Pennings et al. (2017). Pennings et al. (2017) presented a dynamic intermittent demand forecasting method. DLP anticipates the incoming demand of spare parts by including the positive cross-correlation between demand sizes and interarrival times (Haan, 2021; Pinçe et al., 2021). In Pennings et al. (2017), the performance of the method depends on the data set and forecast accuracy measure used. Five different data sets are used (Electro, ElecInd, Raf, Auto and Navy) and two forecast accuracy measures (MASE and GMAE). Most of the time, SBA performs best. SBA outperforms the other methods in terms of MASE and GMAE when applied to the ElecInd and Auto data sets. Furthermore, SBA outperforms the other methods in terms of MASE for the Raf data set and the Navy data set. DLP is the best performer in terms of GMAE for the Electro, Raf and Navy data set. TSB achieved the lowest (being the best performing) MASE for the Electro data set.

2.1.2 Non-parametric approaches

Haan (2021) explains that bootstrapping is used to simulate the distribution of the missing data, by resampling existing data. By doing this, more data is available to model. Bootstrapping was first introduced by Efron (1979). A often used non-parametric bootstrapping approach, that is also used in our comparative study, is the one by Willemain et al. (2004), short WSS. As explained in Pinçe et al. (2021), Willemain et al. (2004) modify the existing bootstrapping method. The new method takes into account three features of intermittent demand: autocorrelation, frequently repeated values and relatively short time series, which were neglected by the classical bootstrapping method. Haan (2021) sums up the detailed explanation given by Willemain et al. (2004). WSS uses a Markov model to first forecast a sequence of zero and non-zero values over lead time periods based on past demand. After this, all the non-zero forecasts obtain specific numerical values. The attributed numerical values are obtained from a random sample of past non-zero values. Lastly, the jittering process starts. The jittering process is explained in detail in Section 3.1.3. It allows to obtain new demand sizes and to smoothen the demand distribution.

Other non-parametric bootstrapping methods are those by Zhou and Viswanathan (2011), Porras and Dekker (2008). Zhou and Viswanathan (2011) is seen as an improvement of WSS. The difference to WSS is, that Zhou and Viswanathan (2011) generate the non-zero lead-time demand by using a bootstrap of the past distribution of the inter-demand intervals. Pinçe et al. (2021) and Haan (2021) explain that Porras and Dekker (2008) is referred to as an empirical method and that it is simpler than bootstrapping.

The last non-parametric method used in this comparative study is quantile regression. Quantile regression as a spare parts demand forecasting method, has not been applied a lot. This is also why it is difficult to find good research papers on it. Trapero et al. (2019) use a quantile combination scheme. First, they obtain the quantiles of the lead time forecast density function. Then, they determine the safety stock. The researchers explain, that based on Boylan, Syntetos, et al. (2006), the whole forecast distribution is not needed. Boylan, Syntetos, et al. (2006) suggest, that only the upper quantiles should be taken into account.

2.1.3 Machine Learning methods

A newer category of methods is ML methods. ML in spare parts demand forecasting is still new and not fully studied. Pinçe et al. (2021) and Haan (2021) explain that ML methods used in the field of spare parts demand forecasting are supervised learning methods. A supervised learning method is an algorithm, that learns from one set of data *(training set)* and then tries to predict the outcome of the other set of the data *(test set)* (Learned-Miller, 2014). Haan (2021) also correctly points out, that ML methods, such as neural networks are often difficult to interpret. In the ML field, researchers speak of a black box, because the input and the output are known, but the way of getting that result is unknown (Rudin, 2019). However, in spare parts demand forecasting, the result is more important (high accuracy and good inventory performance) (Haan, 2021). Another important aspect of ML methods is the hyper parameter tuning part. Makridakis et al. (2022) refers to Makridakis et al. (2018), which state that there are many adjustments possible for the hyper-parameters and that finding the best ones takes time.

One of the first ML methods used in the field of spare parts demand forecasting is MLP [3] by Gutierrez et al. (2008) (Haan, 2021; Spiliotis et al., 2020). MLP is a form of a neural network approach. Gutierrez et al. (2008) find that neural network models generally perform better than the traditional methods (Single exponential smoothing, Croston's method and Syntetos-Boylan approximation) in forecasting lumpy demand. This finding is backed by using three different performance measures (MAPE [4], PB [5] and RGRMSE [6]). Essentially, after Gutierrez et al. (2008)' neural network method, other neural network approaches are introduced. Kourentzes (2013) proposes his version of neural network method, that can handle intermittent time series. In terms of performance accuracy, neural networks perform worse than the best performing Croston's method variation. Whereas, in terms of service level, neural networks achieved better results. One less studied neural network method is LSTM [7]. LSTM was introduced by Hochreiter and Schmidhuber (1997) and has not been extensively studied in the field of spare parts demand forecasting. Chandriah and Naraganahalli (2021)'s paper is one of the few, that use the LSTM method with a modified Adam optimizer for automobile spare parts demand forecasting. Their method is superior to SES, TSB, SBA, Croston and Modified SBA, in terms of forecasting accuracy and inventory management.

Other non-neural networks ML methods, that are used in the spare parts demand forecasting field, are LightGBM and Random Forest [8]. The former was the base of many of the best performing models in the Makridakis et al. (2022) (Haan, 2021). LightGBM is a gradient boosting algorithm. Haan (2021) uses LightGBM in his thesis. LightGBM is the worst performer based on the Percentage Better comparison. However, LightGBM performs well, but not as well

[3] Multi-Layer-Perceptron
[4] Mean Absolute Percentage Error
[5] Percentage Best
[6] Relative Geometric Root-Mean-Square Error
[7] Long Short Term Memory
[8] RF

as MLP for extremely high intermittent data, based on inventory control performance. Overall, SBA seems to be the best performer in Haan (2021)'s paper. The latter is used by Spiliotis et al. (2020). RF is the best performing method in general, before other ML methods, such as Gradient Boosting Trees, MLP, Bayesian Neural Network, K-Nearest Neighbor Regression, Support Vector Regression and Gaussian Processes. It is important to note that the ranking slightly changes depending on the type of data. However, in general, Gradient Boosting Trees and RF perform the best in respect to inventory performance and prediction accuracy. Another paper, that applies RF to forecast spare parts demand is Choi and Suh (2020). They prove that RF is superior to Support Vector Regression, Linear Regression and Neural Network based on MAE [9] and RMSE [10] when applied to South Korean aircraft data.

2.1.4 Comparison between parametric, non-parametric and ML methods

Since the spare parts demand forecasting field is a challenging scientific topic, many new methods are being introduced. Researchers and field experts compare the methods with each other. Based on the reviewed literature, one can say that different outcomes have been found. Willemain et al. (2004) show that, in comparison to SES and Croston's, their method yields a better forecast accuracy of the demand distribution for a fixed lead time. So in this paper, non-parametric methods are superior to parametric approaches. In Spiliotis et al. (2020), ML methods outperform statistical methods, except for SBA, which ranks 7 out of 18 methods. The top three performing methods, based on RMSSE [11] and AMSE [12] for all four types of data, are Support vector regression, Gradient boosting trees and RF (order changing, depending on data type and measure). Pinçe et al. (2021) concludes, that in general, the non-parametric methods outperform the parametric methods. In Pennings et al. (2017), the parametric methods perform much better than the non-parametric methods for the same data, where Lolli et al. (2017) concludes the opposite. It is noteworthy, that both papers use different methods for parametric and non-parametric approaches, which could explain the contradictory results.

2.1.5 Conclusion time series forecasting methods

In summary, one can say that many different time series forecasting methods exist. We presented the three major categories (Parametric, non-parametric, ML methods). Parametric methods, being the first developed methods, are most often used as benchmark methods. Namely, Croston and SBA. Croston, being the first method developed for spare parts demand forecasting. (Cros-

[9] Mean Absolute Error
[10] Root Mean Squared Error
[11] Root Mean Squared Scaled Error
[12] Absolute Mean Scaled Error

ton, 1972). Non-parametric and ML methods have been developed more recently. ML methods, especially, are being studied extensively as they yield promising results in other supply chain management contexts (Pinçe et al., 2021). Next to delivering high forecasting accuracy, solid inventory performance, these methods also need to take into account the type of data. In fact, spare parts data can show extremely high intermittency. Not every method performs well for every forecasting accuracy measure, for every inventory performance measure and for every data set. This literature review suggests, that the different strengths of the different methods should be combined, such that the forecasting accuracy stays high and the improvement in inventory management cuts the costs of stock keeping.

2.2 Performance measures

In the spare parts demand forecasting field, to measure the performance of the methods, two categories of performance measures are mainly used, forecasting accuracy and inventory performance.

2.2.1 Forecasting accuracy

Forecast accuracy measures allow to quantify the performance of the prediction made by the model. It compares a historic value from the training set to the actual value from the prediction. In the case of spare parts demand forecasting, the demand of a spare part is predicted and then compared to the actual value from the test set.

Pinçe et al. (2021) and Haan (2021) explain, that there are two types of forecasting accuracy measures: *relative accuracy measures* and *absolute accuracy measures*. The former quantifies the performance of different forecasting methods relative to each other, while the latter gives an indication of the forecasting error. Haan (2021) and Syntetos and Boylan (2005). Pinçe et al. (2021) show that 72.6% of the papers that they reviewed, use an absolute accuracy measure. Pinçe et al. (2021) provide the table below, which contains commonly used absolute accuracy measures.

Table 1: Common absolute accuracy measures

$ME_t = \frac{1}{t}\sum_{s=1}^{t} e_s$	Mean error (bias)
$MAE_t = \frac{1}{t}\sum_{s=1}^{t} \lvert e_s\rvert$	Mean absolute error
$MSE_t = \frac{1}{t}\sum_{s=1}^{t} e_s^2$	Mean squared error
$RMSE_t = \sqrt{\frac{1}{t}\sum_{s=1}^{t} e_s^2}$	Root mean squared error
$MAPE_t = \frac{\sum_{s=1}^{t} \lvert e_s\rvert}{\sum_{s=1}^{t} Y_s}$	Mean absolute percentage error
$MASE_t = \frac{1}{t}\sum_{s=1}^{t} \frac{\lvert e_s\rvert}{\frac{1}{t-1}\sum_{i=2}^{t}\lvert Y_i - Y_{i-1}\rvert}$	Mean absolute scaled error
$GMAE_t = (\prod_{s=1}^{t} \lvert e_s\rvert)^{\frac{1}{t}}$	Geometric mean absolute error
$GRMSE_t = (\prod_{s=1}^{t} e_s^2)^{\frac{1}{t}}$	Geometric root mean squared error

2.2.2 Inventory control performance

A high forecast accuracy alone does not necessarily mean that the inventory is well managed (Pinçe et al., 2021; Syntetos & Boylan, 2006; Syntetos et al., 2010; Teunter & Duncan, 2009). Therefore, inventory performance measures are also needed, as the cost implications of stock holding are high and even higher for not having a stock at all. In comparison to forecasting accuracy measures, inventory control measures do not compare the difference between the mean and the forecast. Inventory control measures measure the effectiveness of the stock management in terms of achieved cycle service level, trade-off curve, total cost, stock volume or shortage volume. Therefore, a distribution of the demand needs to be assumed. In Section 3.3, the choice of the assumed distribution for our paper is explained. Pinçe et al. (2021) provide a visualisation of the inventory performance measures used in spare parts demand forecasting, where *Service level* and *Trade-off curve* are the two most used measures in front of *Total cost, Other (Average total cost, Average on-hand inventory or Stock-out volumes), Stock Volume and Shortage Volume*. Including inventory control measures allows to see the financial implications of inventory management.

2.3 Comparative studies

Finally, the third major category of spare parts forecasting is *Comparative studies*. Different spare parts forecasting methods will be benchmarked and compared to each other after being applied to the different data sets. The methods used for this comparison study will be presented in Section 3.1. The results of each method applied to each data set are then quantified to be able to compare their performances (Pinçe et al., 2021). The performance measures used are forecasting accuracy measures or inventory performance measures. Both types of measures will be used in this paper. Pinçe, Turrini, and Meissner explain that most studies use forecast accuracy measures, as there seems not to be a general convention on which methods to use as a benchmark. However, inventory performance is described to provide more realistic benchmarks in Teunter and Duncan (2009). A recent paper that takes both into account is Haan (2021). This step is important for the field of spare parts demand forecasting as it allows to constantly challenge the findings of researchers and compare them with each other. Furthermore, in the last five years, there have not been many new comparative studies (Pinçe et al., 2021).

More recent comparative studies are the Master thesis of Haan (2021) and the paper of Aktepe et al. (2021). To our knowledge, the newest paper that compares spare parts demand forecasting methods, are İfraz et al. (2023) and Theodorou et al. (2023).

Haan (2021) compares seven methods with each other. Five conventional methods: Simple exponential smoothing (SES), Croston's method, Syntetos-Boylan approximation (SBA), Teunter-Syntetos-Boylan (TSB) and Willemain and two ML methods: Multi-Layer-Perceptron (MLP) and LightGBM. The methods are applied to the same eight data sets (four industrial data sets and four simulated data sets), that are used for our paper. De Haan concludes that based on the Percentage Better [13] comparison, SBA performs best overall and LightGBM performs worst. This relative measure allows to determine the superior methods. When comparing the performances based on inventory control performance, Willemain is the best performing method. This is only true for data that is not categorized as extremely intermittent. For such demand, Haan (2021) concludes, that MLP and LightGBM are the best performer. Two critiques of this paper are, that De Haan includes the TSB method, although obsolescence is not identified and that the LightGBM model is not tuned for the hyper parameters. In fact, for the latter, De Haan relies on the parameter values of Kailex (2020). Obsolescence is explained as a spare part no longer being needed, which means, that the demand for that item goes towards zero (Van Jaarsveld & Dekker, 2011). TSB was introduced in 2011 by Teunter et al. (2011) as an improvement to Croston's method, as the latter yields poor performance for obsolescence. Nonetheless, the obsolescence can be implicit and unidentified.

İfraz et al. (2023) compare four different types of methods for spare parts demand forecasting. The types of methods used are: Regression-based methods (multivariate linear regression[14] multivariate nonlinear regression [15], Gaussian process regression [16], additive regression [17], regression by discretion [18], support vector regression [19]), Rule-based methods (Decision table,

[13]PB
[14]MLR
[15]MNR
[16]GPR
[17]AR
[18]RbD
[19]SVR

M5Rule), Tree-based methods (Random Forest [20], M5P, Random tree, Reduced Error Pruning Tree [21]) and Artificial neural networks [22]. This paper's contribution is important to the spare parts demand forecasting field, as it uses more ML methods than previous comparison studies. The researchers apply the methods to a data set of an urban transport bus fleet of a metropolitan municipality. The inventory type is classified using an Always Better Control method (ABC). The ABC method follows two rules. The first rule states that items of low value should be amply kept in stock. The second rule dictates that the quantity of the items of high value should be sparse, but should be checked more frequently. Although the ABC classifier method is not used in our paper, İfraz et al. (2023) include multiple ML methods, which provide guidance for our paper, as we are applying some of the ML methods to our data sets.

In Aktepe et al. (2021), four methods (Linear regression [23], Nonlinear regression [24], ANN and SVR) are used to predict the sales of a construction machinery company. Its business consists of the sale of spare parts it produces for other companies. The researchers explain in the *Conclusion and Discussions* part, that the ML methods are performing better than the linear and nonlinear regression models in terms of forecasting accuracy. This is also a reason to analyze the performance of ML methods in our paper, as ML methods look promising in the field of spare parts demand forecasting. Nonetheless, they do not provide inventory performance measures, which would allow to observe if the findings stay consistent. This is why, in our paper, next to the forecast accuracy measures, inventory control measures are used to test for a difference in the outcome.

Most recently, Theodorou et al. (2023) conducted a study on the connection between forecasting accuracy methods and inventory performance methods applied to the M5 competition data set from Makridakis et al. (2022). The inventory performance measures used are trade-off curves and monetary cost estimates (lost sales and holding inventory), as the cost variable is available in the data set. 12 forecasting accuracy methods are used in this paper:

- Naive & seasonal Naive (sNaive) methods
- Moving Average (MA)
- Simple Exponential Smoothing (SES)
- Croston
- Syntetos-Boylan Approximation (SBA)
- Teunter-Syntetos-Babai (TSB)
- Automated selection of exponential smoothing models (ES)

[20] RF
[21] REPT
[22] ANN
[23] LR
[24] NLR

- Automated selection of ARIMA models (AutoRegressive Integrated Moving Average)
- Aggregate-Disaggregate Intermittent Demand Approach (ADIDA) & intermittent Multiple Aggregation Prediction Algorithm (iMAPA)
- LightGBM

To measure the accuracy of the forecasts, Root Mean Squared Scaled Error (RMSSE) is used. The performance of the models is related to the length of the review period. The ranking of the performances of the methods is provided in Table 2. The researchers conclude, that the optimal choice of forecasting method may vary depending on the assumed costs. Furthermore, the choice of forecasting method should be connected to the target, as more accurate methods do not necessarily show lower costs. Only one forecasting accuracy measure, namely RMSSE is used in this paper, which does not allow to compare the performances of the methods based on the choice of forecasting accuracy measure. This is why our paper uses next to RMSSE also MSE and MASE. Regarding the inventory performance, the researchers assume a normal distribution, which differs from our case, as we assume a gamma distribution. Nguyen (2023) compared the normal and gamma distribution for our data sets and concluded, that the gamma distribution performs better.

The research question and the sub-questions of our paper are also of great importance in Pinçe et al. (2021). For the former, there is no simple answer to it. Pinçe et al. (2021) explain that the performances of the methods vary from one industrial data set to another. For the latter, the inventory performance measure and accuracy measure used play a big role in the results as they can yield different outcomes. Furthermore, the hand-in-hand use of inventory performance measures and accuracy measures is advised as both do not necessarily show the same performance results. Regarding the sub-question about the performance of ML methods compared to the performance of statistical methods, Pinçe et al. (2021) explain that, as mentioned in Baryannis et al. (2019) and later by Kraus et al. (2020), ML methods are of good use in other supply chain management contexts. This is why, they could work better in spare parts demand forecasting.

From this literature review, we conclude that, as there have not been a lot of comparative studies in the spare parts forecasting field lately, this paper can contribute to this field. Furthermore, we can conclude that ML methods have a lot of potential in the spare parts demand forecasting field, as in two out of the four reviewed comparative papers, they perform better than traditional methods. Nonetheless, this finding should not be trusted blindly. In fact, out of the four comparative papers, only Haan (2021) uses multiple data sets. The other papers only apply their methods to one single data set. Furthermore, when comparing the performances of the methods, a lot of variability regarding the superior method is observed. In fact, the findings of the papers, that use some of the same methods are not consistent with each other. Our research question seems to be important in other papers too. The literature review provides guidance on how other researchers approached the research questions. This can be taken into consideration for this paper, however, also focusing on different forecasting methods.

Thus, the following methods are the most promising methods for our research: Croston, SBA, MLP, Willemain, RF and LightGBM. Furthermore, as performance measures, we decide on MSE, MASE, RMSSE and Trade-off curves. We also decided to use other methods and measures that have not been used in the comparative papers from Table 2. Those methods are DLP, Quantile regression and LSTM for forecasting and GMAE to measure the accuracy. The reasons why we use those methods and measures are given in Section 3.1.

Table 2 and 3 on the next two pages, provide an overview of the literature review of recent comparative studies in the spare parts demand forecasting field and their key findings.

Table 2: This table gives an overview of the data, the methods, and the performance of the comparative papers.

Paper	**Data**	**Methods**	**Forecasting Accuracy**	**Inventory Performance**
Haan (2021)	MAN BRAF AUTO OIL SIM1 SIM2 SIM3 SIM4	Croston SES SBA TSB Willemain MLP LightGBM	MSE MASE RMSSE	Trade-off curves Service levels
İfraz et al. (2023)	Urban bus fleet	MLR MNR GPR AR RbD SVR Decision table M5Rule RF M5P Random tree REPT ANN	MAPE	None
Aktepe et al. (2021)	Construction machinery company in Turkey	LR NLR ANN SVR	MAPE	None
Theodorou et al. (2023)	M5 competition by Makridakis et al. (2022)	Naive& sNaive MA SES Croston SBA TSB ES ARIMA ADIDA iMAPA LightGBM	RMSSE	Trade-off curves *(Normal distribution assumed for the target service level)* Monetary cost

Table 3: This table summarizes the key findings of the reviewed comparative studies.

Paper	Key Findings
Haan (2021)	According to the *Percentage Better* comparison, SBA performs best overall. LightGBM performs worst. Based on inventory control performance, Willemain's method performance is higher than other methods, except for high intermittency demand data, where MLP and LightGBM seem to perform best for the used inventory control measures.
İfraz et al. (2023)	ANN outperforms every method. Decision Tree performs better than M5Rule in the category of rule-based methods. SVR is the best performing method out of the regression-based methods. Out of the tree-based methods category, M5P has the lowest MAPE. Comparing all four categories, ANN performs best, followed by rule-based methods, regression-based methods and tree-based methods.
Aktepe et al. (2021)	SVR is the best performing method. ANN outperforms Nonlinear regression and Linear regression in terms of MAPE and deviation (in pieces).
Theodorou et al. (2023)	In terms of accuracy, the researcher differentiate based on the length of the forecast horizon ($R \in 1, 3, 7, 14$). The best performing methods through forecast horizons 1 to 7, are ADIDA and iMAPA. LightGBM performs best for a forecast horizon of 14, followed by ADIDA and iMAPA. Naive and sNaive are the worst performer throughout all review periods. Croston's method and its variants highest performance is the 5th best performance out of the 12 methods for R = 1. Based on inventory performance, the findings are generally in line with the forecasting accuracy findings. However, ARIMA seems to stand out with LightGBM for the higher review periods.

3 Research design and methodology

This paper will be structured in two steps. The first step consists of setting up the experimental design based on previous comparative studies like Haan (2021). The second step requires the selection of several spare parts forecasting methods and a description of the technique of each method. Furthermore, the measures of the results for the forecasting accuracy and the inventory performance will be elaborated. By using different performance measures on the different applied methods, we aim to be able to answer the research questions and analyse which methods perform best for which performance measures.

In the experimental design, four industrial data sets and the four simulated data sets will be explored and classified in one of the four demand classification categories. This allows us to investigate whether certain methods perform better for certain types of demands.

The data sets have already been cleaned by de Haan (2021) and improved by Nguyen (2023) for efficiency. It is also noteworthy, that outliers have already been removed by the two mentioned authors. Next, the data will be split into a train and test set. Then, the chosen methods will be applied to the data sets, which will then allow us to compare the results of the forecasting accuracy, the inventory performance and the differences in the results due to the different data sets. Further explanations about why those data sets were picked and what their different characteristics are, will be explained in Section 4.

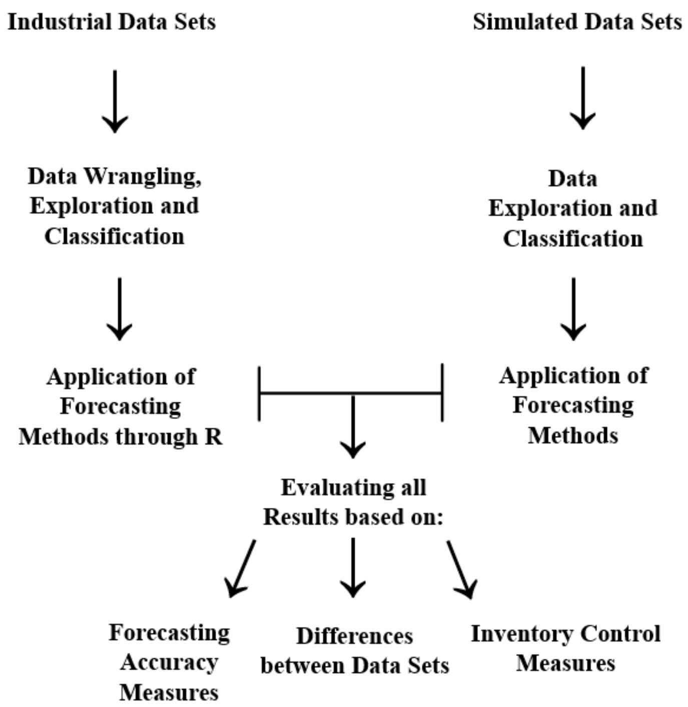


Figure 1: The flow of the experimental design by Haan (2021).

3.1 Methods for the comparison

The methods used can be grouped into three categories:

1. Category: Statistical methods
2. Category: ML methods
3. Category: Non-parametric methods

In the next step, we explain the reason behind using this method in this comparative study and how the method works.

3.1.1 Statistical methods

Let's start with presenting the methods from category 1, that will be used.

Croston

One method, that is commonly used as a benchmark method is the *Croston's method*, which is also the first method developed for spare parts demand forecasting. Croston's method is elaborated in detail in Croston (1972) and is built on the Simple Exponential Smoothing method (SES). Kourentzes (2013) explains that Croston's method focuses on two separate components. z_t, which is the non-zero demand size and x_t, which stands for the inter-demand interval. SES uses a smoothing parameter which puts more weight on the recent data (demand). However, for intermittent demand where zero demand periods can happen, SES would take into account the zero demand periods, which are extreme values that have an impact on the prediction. This is why in Croston's method, z_t has to be non-zero as the estimates are only updated when demand occurs. The prediction of Croston's method is given by: $\hat{Y}_t = \frac{\hat{z}_t}{\hat{x}_t}$. Croston's method is included in this paper, as it allows us to compare the newer methods to Croston's method, a method which is used as a benchmark in most spare parts demand forecasting papers.

Syntetos-Boylan approximation

Another statistical method, serving as a benchmark, is Syntetos and Boylan (2005)'s method also abbreviated as SBA. The decision to include SBA is due to the fact that SBA was developed to prove that Croston's method is biased. SBA corrects the bias. SBA proposes following estimator: $Y_t' = (1 - \frac{\alpha}{2})\frac{z_t'}{x_t'}$, where $(1 - \frac{\alpha}{2})$ is the bias correction coefficient. α represents the smoothing constant value, which is utilized to update the inter-demand intervals. Both Croston's methods and SBA can be applied through the use of the *"tsintermittent"* R-package by Kourentzes (2014).

DLP

The last statistical method is presented by Pennings et al. (2017). The DLP method is an intermittent demand forecasting method and assumes a dependence between interarrival time (elapsed time) and demand size to anticipate the incoming demand, which is not the case in methods like Croston's (Pinçe et al., 2021).

The simplified formula of the DLP method is:

$$D_{L,t} = \mu[L + (\tau_0 - \frac{1-p}{p})(1-(1-p)^L)]$$

The left part of the equation represents the expected total demand $D_{L,t}$ at time period t for an SKU for a lead time of L. The expected total demand is calculated proportionally to the inter-arrival time (τ_0) with respect to the probability of non-zero demand (p). In other words, the DLP method exploits the elapsed time (τ_0) to anticipate incoming demand. This part of the equation: $(\tau_0 - \frac{1-p}{p})(1-(1-p)^L)$ takes into account this elapsed time and adjusts the expected demand by the probability of non-zero demand (p). $1-(1-p)^L$ represents the probability of at least one demand occurring over the lead time period.

The reason for including this method is because Pennings et al. (2017) obtain encouraging results. In fact, the researchers state that they are able to reduce unnecessary inventory investment by 14% for SKUs that exhibit cross-correlation, compared to Croston's method. As no package exists for the DLP method, some code has been provided by Dr Jan Van Dalen (one of the three researchers, that introduced the method). The code is run on RStudio and will be provided on a GitHub page, linked in subsection 6.6, dedicated to the master thesis.

3.1.2 Machine Learning methods

The methods in category 2 are the following ones.

LightGBM

The first method is LightGBM. As described in Haan (2021), LightGBM was the base of many of the top methods in the M5 competition analyzed by Makridakis et al. (2022). Gradient Boosting methods are also generally used for many *Kaggle competitions*, as they perform quite well and are easy to use. This is also the reason, why it will be used in this paper. For this method, the code from de Haan (2021) will be used. However, we will tune the hyper parameters differently, to see if the findings improve. We set the learning_rate to 0.01 (previously 0.O75), increased the number of rounds to 15000 (previously 12000) and got rid of the sub_feature and sub_row arguments. De Haan adapted the code for the same data sets that are used in this paper from Kailex (2020) and did not try other values for the best hyper parameters. The functioning of LightGBM can be found in the open source documentation Microsoft (2021) and the R package used in this paper in Shi et al. (2022). Furthermore, the hyper parameters and their respective roles can be found in Table 4. The description of the roles of the hyper parameters have been obtained on the Microsoft (2021) page.

Long short-term memory

The next method is Long short-term memory (LSTM), which is a type of Recurrent Neural Networks (RNN). LSTM has been proposed by Chandriah and Naraganahalli (2021) to forecast automobile spare parts demand. As explained in Chandriah and Naraganahalli (2021), the difference between an RNN and a feed-forward Neural Network is that the RNN uses a feedback connection to remember the prior time steps. This whole process is quite complicated in the long term, which is why the function of LSTM comes in handy. The latter is able to resolve the problem of vanishing gradient in RNN. The vanishing gradient problem means that with every parameter update, the gradient becomes smaller. However, the gradient is carrying the information. This means that a smaller gradient provides also less information. For long data sequences, this becomes a problem as the updates of the parameter are not significant anymore. In other words, there is no learning happening anymore. In this paper, LSTM will be used in combination with Adam optimizer (Adaptive moment estimation) as in (Chandriah & Naraganahalli, 2021). RNN functions by remembering the output of the previous data point and re-using it for the next one (memory). The Adam algorithm allows to optimize the weights at each level.

The technicalities of RNN and LSTM are explained in depth in Sherstinsky (2020). Or in the seminal paper of Hochreiter and Schmidhuber (1997). I decided to include this method for the comparison, as in the paper of Chandriah and Naraganahalli (2021), the researchers state, that, the modified Adam optimizer performs well for their data set. Furthermore, one of the data sets used in our study is also an automotive data set. However, Chandriah and Naraganahalli (2021)'s paper states that, *"The Croston method forecasts the demand by separating the time intervals and demand size. This method is better compared to conventional Simple Exponential Smoothing (SES), Syntetos-Boylan-Approximation (SBA), Croston, Teunter-Syntetos-Babai (TSB) and Modified SBA. However, the performance of these methods is poor for intermittent demand."*. This is not in accordance with findings from other papers such as Pinçe et al. (2021), Teunter et al. (2011) and other renowned papers. In fact, the TSB method was introduced to adjust for the lagging update of the variation of the new demand levels. Thus, TSB should perform better than the previous methods (Croston and Croston's modifications) for intermittent demand spare parts forecasting in most cases (Pinçe et al., 2021). Furthermore, although Chandriah and Naraganahalli (2021) categorize this paper in the spare parts demand forecasting field, the data consists of new cars and not spare parts. This is different from spare parts demand forecasting and is not helpful for our paper. However, the paper guides us on how to apply the LSTM method.

To run the model, the "keras" and "tensorflow" packages in R are used and the optimizer is set to "optimizer_adam". Furthermore, the *learning_rate*, *beta_1* and *beta_2* can be tuned. keras-team (2021) provides insights into the implementation of the Adam optimizer. Tunable hyper parameters, specifically to the Adam optimizer are the *learning_rate*,β_1, β_2 and *epsilon*. The *learning_rate* controls the step size of the weight updates. β_1 and β_2, represent the exponential decay rate for the 1st moment estimates and 2nd moment estimates, respectively. Simply put, these hyper parameters control how much the optimizer "remembers" its previous moments. *epsilon*, a small constant for numerical stability. In addition to these hyper parameters, there are also hyper parameters to the LSTM model itself, such as the number of layers, the number of units in each layer, the batch size and the number of epochs. Table 4 shows the parameters that can be tuned in LSTM and what their role is. The description of the roles of the hyper parameters have been obtained on the dedicated GitHub page of SciKit-Learn (2015). Further details of the Adam optimizer can be found in the seminal paper of Kingma and Ba (2014). Furthermore, the LSTM method needs some data pre-processing, i.e. setting: "lag", "delay" and "n" (next steps) to obtain the input sequences (X) and output sequences (Y). The "lag" allows to set the number of previous time steps to use as input variables per sequence to predict the next time period. "delay' allows to set the step how far the model will predict into the future. And "n" allows us to set how many time steps ahead the model will predict. The downside of the LSTM method is that due to the need to create sequences, the data becomes scarcer, as the time series data is combined into smaller chunks. This causes an issue in our case, as when the data is split into a training and test set, the test set has fewer time steps, which are used to create the sequences. Hence, there are even fewer predictions generated out of those test data sequences. This is why, when computing the accuracy measures, the test data input is shortened, such that its length matches the length of the predictions. Regarding the hyper parameter tuning, we decided to keep the model simple, i.e. two layers with 50 units each, a dropout layer and 10 epochs to prevent overfitting, as training a model on not much data risks overfitting.

Multi-layer perceptron

Another ML method is the *Feed-forward neural network*, which is based on the methodology of Spiliotis et al. (2020) and can also be referred to as a Multi-Layer Perceptron (MLP). This neural network consists of a single hidden layer. As Haan (2021) mentions from Smyl (2020), all the ML methods are trained the same way, which is using constant size, rolling input and output windows. Haan (2021) and Spiliotis et al. (2020) cite Zhang et al. (1998), which states that because of the use of nonlinear activation functions by ML algorithms, the data should be scaled in the range of 0 and 1 pre-training. By scaling the data, not only does the learning speed improve, but also computational problems are avoided. The data should be linearly transformed between 0 and 1 following $y' = \frac{y_t - y_{min}}{y_{max} - y_{min}}$. The transformation is reversed after obtaining the forecasts, to find out the final prediction and the forecasting accuracy. This method is included in this comparative study, as it is easy to run, yet performs well in other papers. Furthermore, it allows us to observe the performance of simple ML methods compared to statistical spare parts forecasting methods. To run this method, the *RSNNS* package in R will be used and the hyper parameters will be tuned until the optimal parameters are found for the training of the model. Table 4 shows the parameters that can be tuned in MLP and what their role is. The description of the roles of the hyper parameters have been obtained on the dedicated GitHub page of SciKit-Learn (2019b).

Random forest

The last ML method is based on the Random forest algorithm proposed by Breiman (2001). Random Forest combines the predictions of multiple decision trees and averages their predictions (Biau & Scornet, 2016). Spiliotis et al. (2020) used RF in their comparative study and implemented it by using the R package *randomForest* by Liaw, Wiener, et al. (2002). We decided to include this method, as Random Forest is easy to apply. Furthermore, Choi and Suh (2020) compare Random forest in their paper to *Support Vector Regression, Linear Regression and Neural Network*. Random Forest yields the best results in their paper. The Random Forest algorithm allows to tune several hyper parameters, which can be seen in Table 4. The description of the roles of the hyper parameters have been obtained on the dedicated GitHub page of SciKit-Learn (2019a) and in the paper of Probst et al. (2019). Throughout the implementation of RF, several problems came to our attention. One problem is, that it is highly computational intensive. This is why the method is run on Google Colab in an R script. The model itself is built on the *scorecardModelUtils* and *randomForest* packages by Arya Poddar (2019) and Liaw, Wiener, et al. (2002). We decided to use the *scorecardModelUtils* and *randomForest* packages for Random Forest, as the former package allows hyper parameter tuning and the latter package is used to train the final model and the prediction.

It is important to know, that many hyper parameters can be tuned for ML methods. Not every single tuneable hyper parameter is shown in Table 4, as this is beyond the scope of this thesis. Furthermore, the used package also plays an important role for the used hyper parameters, as some hyper parameters cannot be tuned in some packages.

The ML methods are expected to perform well. However, the hyper parameter tuning of those methods will be an important part and the most difficult part of implementing ML methods. By correctly tuning the model, the methods can be reproduced by others, which allows standardization of the procedure. Another important point regarding ML methods is, as already mentioned in Section 2.1.3, the lack of interpretability. The so-called *Black box* problem can occur for some ML methods, that use complex mathematical operations and data transformations. In our case, the *Black box* problem is mainly an issue for the MLP and LSTM methods as these are Deep learning methods. Deep learning is a subset of ML, which requires more amount of data and a longer training time. Although it requires less human intervention, as Deep learning methods learn on their own, they make non-linear, complex correlations, which are difficult to understand. LightGBM and RF, on the other hand, are easier to interpret, as they are tree-based methods, that can be visualized.

Table 4: This table summarizes the important hyper parameters of the used ML methods and their roles.

Parameter	**Light GBM**	**MLP**	**LSTM**	**RF**	**Role**
Nr. of leaves	**X**				Controls the complexity of the tree model.
Min. data in leaf	**X**				Prevents over-fitting.
Max depth	**X**			**X**	Limits the tree depth explicitly. (Not for RF in R (only Python).
Objective	**X**				Specifies the application.
Boosting	**X**				Specifies the type of boosting algorithm to use.
Learning rate	**X**	**X**	**X**		Controls the step size in updating the weights.
Feature fraction	**X**				Controls the percentage of features used at the beginning of each tree.
Bagging fraction	**X**				Specifies the fraction of the randomly selected data for use in each training iteration.
Bagging freq.	**X**				Specifies the frequency for bagging.
$\lambda L1$	**X**				Adds a penalty term to the loss function.
Nr. of hidden layers		**X**	**X**		Controls the complexity of the model.
Nr. of neurons per layer		**X**	**X**		Controls the capacity of the model.
Activation func.		**X**	**X**		Specifies the activation function.
Optimizer		**X**	**X**		Controls the solver for weight optimization.
Alpha		**X**			Controls the strength of the L2 regularization term.
Batch size		**X**	**X**		Controls the size of the minibatches for stochastic optimizers.
Nr. of LSTM units per layer			**X**		Controls the nr. of LSTM units in each hidden layer.
Dropout rate			**X**		Controls the amount of regularization applied.
Nr. of trees				**X**	Nr. of trees in the forest.
Splitting rule				**X**	Splitting criteria in the nodes.
mtry				**X**	Number of drawn candidate variables in each split.
node size				**X**	Minimum number of observations in a terminal node.

3.1.3 Non-parametric methods

Willemain

Willemain et al. (2004) method is different from the statistical and ML forecasting methods because they forecast a whole distribution of demand over a fixed lead time. It does this, by using a bootstrap method in 7 steps, they are able to forecast the cumulative distribution of demand over a fixed lead time. Willemain's method can be summarised in 7 steps:

1. Step: Estimate transition probabilities for two-state Markov model for historical demand
2. Step: Utilize the Markov model to generate zero and nonzero sequences over the forecast horizon conditional on the last observed demand.
3. Step: Replace nonzero demand with a random numerical value with replacement, from the set of observed nonzero demands.
4. Step: Jitter the nonzero demand values. *Jittering* means to pick a different value, which is located close to the selected value. This allows more variation and a more natural variation of the demand size. (Example: Instead of using the randomly chosen non-zero demand of 7, a close-by value such as 6, 8, 9 or 10 is used.) The maximum value is the previous value plus the jittering value.
5. Step: Summation of the predicted values over the forecast horizon, to get one single predicted value of lead-time demand (LTD).
6. Step: Repeat steps 2-5 many times to obtain many LTD values.
7. Step: The obtained LTD values in step 6 are sorted, such that a distribution of LTD is obtained.

The lead time for Willemain is set to 1 instead of 0 because a lead time of zero would mean that only the current period is being forecasted. This is because Willemain's bootstrapping method forecasts a cumulative distribution of the demand over a certain lead time. If the lead time is 0, there is no delay between the decision to replenish and when the stock is available. This means that as there is no delay to account for, the method is forecasting the demand for the current period. As Haan (2021) mentions that Willemain et al. (2004) successfully proves that his method outperforms SES and the methods based on SES, such as Croston's method, Willemain's bootstrapping method is also included. However, Willemain's method has a critique point. In fact, when sampling for one single period ahead, the Markov chain is reduced as it can only have one of the two states, zero demand or non-zero demand. To run the method, the code from Nguyen (2023) is used.

Quantile regression

The last method, quantile regression, is also categorised as a distribution-focused method as the quantile function is the inverse of the distribution function (Taylor, 2007). Furthermore, quantile regression estimates the conditional quantile function as a linear combination of the predictors and does not make assumptions about the distribution of the target variable. Koenker and Hallock (2001) explain that quantile regression is suited for cases when the conditions of linear regression are not met (i.e. linearity, homoscedasticity, independence and normality). In the spare parts forecasting domain, quantile regression would allow to find a specific quantile that suggests, for example, taking the 25th quantile, there is a 25% chance that the actual demand for a spare part is below the forecast and there is a 75% chance that the demand is above. The quantile regression model is given by: $Q_\tau(y_i) = \beta_0(\tau) + \beta_1(\tau)x_{i1} + ... + \beta_p(\tau)x_{ip}\ i = 1, ..., n$ and $\tau \in (0, 1)$. $Q_\tau(y_i)$ represents the τ-th quantile of the dependent variable 'y' for the i-th observation. $\beta_0(\tau)$, $\beta_1(\tau)$, $\beta_2(\tau)$, ..., $\beta_p(\tau)$ are the quantile-specific coefficients of the intercept and independent variables at the τ-th quantile. x_{ip} are the independent variables for item i and period p. τ represents the quantile level of interest. For every wanted quantile, in our case from 50% to 99%, we fit a quantile regression for every period to predict the next demand 'y' based on the previous predictions.

As we are focusing more on the upper quantiles (i.e. from 50% to 99%), the values need to be converted into percentages. For this, no extra package is needed. In fact, after establishing the quantile regression model with the existing *rq* function in R, predictions for the desired quantiles can be made through the *predict* function, by setting the *'level'* argument to a vector of desired values. The rq function takes as input the formula, the data and the τ (quantiles) levels of interest. In our case, general insights into the overall performance of the model across the pre-determined range of quantiles can be made through this. Quantile regression is included in this comparative study, as there are not many papers that use this method for spare parts forecasting (Syntetos et al., 2012).

3.2 Selected forecasting accuracy measures

After training the model, the model needs to be tested. Therefore, accuracy measures are required, that allow us to compare the predictions with the actual values. For this, the most, widely used accuracy measures are used (Pinçe et al., 2021). As Haan (2021) describes from Pinçe et al. (2021), the most commonly used accuracy measures are *absolute accuracy measures*. In this comparative study, Mean Absolute Scaled Error (MASE) is one of the absolute accuracy measures. MASE is quite important as it allows a scale-free measurement across all time series of different items (Pinçe et al., 2021). The other absolute accuracy measure is Mean Squared Error (MSE), which has been proposed by Hyndman and Koehler (2006). These accuracy measures are defined respectively as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(Y_t' - Y_t)^2, MASE = \frac{\frac{1}{n}\sum_{t=1}^{n}|Y_t' - Y_t|}{(\frac{1}{n_1-1})\sum_{i=2}^{n_1}|Y_i - Y_{i-1}|}$$

The third accuracy measure is the Root Mean Squared Scaled Error (RMSSE), which has been elaborated by Hyndman and Koehler (2006) and used in many papers, such as Haan (2021), Spiliotis et al. (2020), and Theodorou et al. (2023). Again, as the MASE measure, the RMSSE also allows a scale-free measurement across all time series of different items. It is defined as follows:

$$RMSSE = \sqrt{\frac{\frac{1}{h}\sum_{t=n+1}^{n+h}(y_t - \hat{y}_t)^2}{\frac{1}{n-1}\sum_{t-2}^{n}(y_t - y_{t-1})^2}}$$

Theodorou et al. (2023) describe from Kolassa (2016), that Squared errors measures, such as RMSSE, are suitable when it comes to estimating the average demand for intermittent data.

Last but not least, the less used Geometric Mean Absolute Error (GMAE) is the fourth applied absolute accuracy measure in this thesis. GMAE is used in the paper of Pennings et al. (2017) next to MASE. Pennings et al. (2017) claim that GMAE and MASE are two recently proposed and widely used metrics. The former does not scale the errors, whereas the latter does. As we are also using the DLP method, we want to be able to compare the performance of the method with Pennings et al. (2017)'s results. GMAE is defined as follows:

$$GMAE = (\prod_{s=1}^{t}|e_s|)^{\frac{1}{t}}$$

e_s is the prediction error, between the actual value (demand in our case) and the predicted value. To obtain the GMAE, the absolute errors of each observation are multiplied by each other. After that, the t-th root of this product is taken. t being the total number of observations. However, the GMAE is not well-suited for data containing a lot of zeros. This goes for the data set containing the predictions and the test data set. In fact, in the context of GMAE, if the predicted value is zero and the actual value is also zero, then the absolute error for that prediction is zero. Since GMAE is the geometric mean of these absolute errors, if any of the absolute errors are zero, then the GMAE will be zero. The Table 5 below provides a small example of the sensitivity of GMAE for 0 values.

Predicted	Actual	Absolute error
4	3	$\|3-4\|=1$
0	1	$\|1-0\|=1$
0	0	$\|0-0\|=0$
Geometric mean	$(1*1*0)^{\frac{1}{3}}=0$	
t = the total number of observations		

Table 5: Example of GMAE with a 0 value as absolute error.

Now, to be able to compare the performance of the accuracy measure of the different methods on different data sets, a new measure is needed. Pinçe et al. (2021) use the *Percentage Better* (PB) and *Percentage Best* (PBt) and explain that both "rank the performance of different methods based on the percentage of time they perform better or best according to an underlying measure.". PB and PBt are relative accuracy measures. Given that Haan (2021) uses PB, we are also going to use PB as this allows us to compare our findings to their findings on the same data sets.

3.3 Selected inventory control measures

As already mentioned in Section 2.3 Comparative studies, two types of performance measures are used. Next to the forecasting accuracy measures, inventory performance measures are also important, as the former does not necessarily mean that the inventory performance for spare parts is high. While most forecasting methods estimate the mean, inventory control measures need an assumption of the demand distribution. In our case, we rely on Nguyen (2023)'s findings, which show that a gamma distribution performs better than a normal distribution. However, this is only the case for when the mean is not too small compared to the variance. In fact, a company prefers to have too much stock rather than too little, as it can then at least minimize downtime. This means, that the loss function is considered to be asymmetrical. Pinçe et al. (2021) present a distribution plot (Figure 5 on page 13) that shows the two most commonly used inventory performance measures are the *Service level* and the *Trade-off curve*. In our paper, as in Haan (2021)'s paper, the trade-off curves show the trade-off between the achieved fill rate (AFR) and the holding costs.

Before determining the AFR, an inventory policy needs to be set (Haan, 2021). In this paper, the approach by Van Wingerden et al. (2014), which is also used by Haan (2021) is used. Herefore, a base stock level R is determined by evaluating previous demand. Each period, the *Inventory Position* (IP) is updated. Back ordering is allowed. IP is defined as:

$$IP = stock\ on\ hand + outstanding\ orders - back\ orders$$

Van Wingerden et al. (2014) state, that if IP drops to the stock level (R) or below, new stock is ordered. Although a minimum order quantity can be specified, we rely on Haan (2021)'s paper and decide to also not include it, for simplicity reasons. Furthermore, a zero lead time is assumed. Zero lead time indicates that the replenishment order arrives immediately after an order is placed.

By picking the same *Inventory control measures* as Haan (2021), the comparison of the inventory performance of the same data sets used with different methods across different comparative studies is possible. As in this paper, the trade-off curves visualize the trade-off between the AFR and the holding cost, a target fill rate (TFR) needs to be set. The fill rate targets used for this paper are 75%, 80%, 85%, 90%, 95%, 99% and 99,9999%, which are the same as Haan (2021) used.

3.4 Demand classification and data training

Before training a model, the data is split into a training set and a test set. For this, we will apply the same training procedure as Haan (2021), which is shown in Nguyen (2023). The data is split into a 70% and 30% split. This means, that 70% of the data are used for training the model and the other 30% for testing the model, to see how accurate the model is. The training is done on a single SKU [25] basis.

The industrial data sets need to be classified first into one of the four categories: *Erratic, Lumpy, Smooth and Intermittent*. The classification is done by respecting the classification scheme of Boylan et al. (2008), which is based on Syntetos and Boylan (2005). Boylan et al. (2008) suggest that the classification is done based on two criteria, the mean inter-demand interval p and CV^2, the squared coefficient of variation of the demand sizes.

The mean inter-demand interval "p" for every item $= \frac{\text{Total number of time periods}}{\text{Count of the non zero demands}}$ and

$$CV^2 = \left(\frac{\text{Standard deviation of the non zero demands}}{\text{Mean of the non zero demands}}\right)^2$$

[25] Stock Keeping Unit

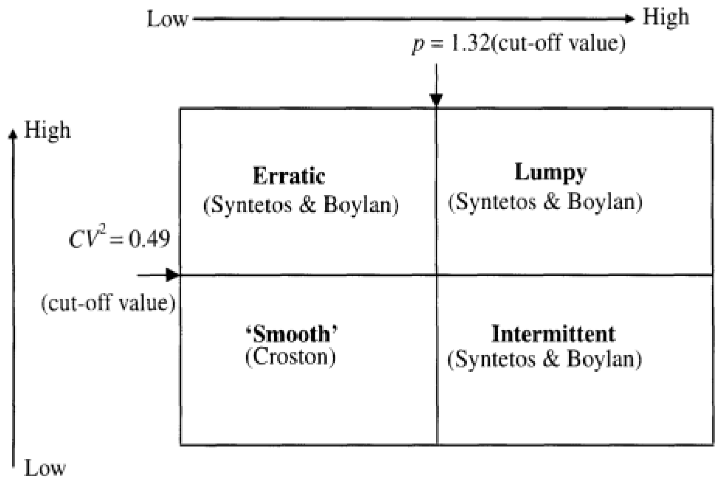


Figure 2: Demand-based categorization for forecasting by Boylan et al. (2008).

4 Data

The data sets for this paper are divided into four industrial data sets and four simulated data sets. The reason behind using industrial data sets and simulated data sets is, that the latter allow to control the environment. This means that by having a complete *intermittent* demand data set, this allows to see the impact of the demand class on the method, which allows us to answer the main research question of this study. Including industrial data, allows to observe the reality, how the methods perform in practice when used in an industry. All data sets can be found on the GitHub page of Nguyen (2023). The data sets have been cleaned and the outliers have been removed by Nguyen (2023). An important aspect of the data sets is whether they include lead time or not (Haneveld & Teunter, 1997). In case, lead time is not included in the data set, a lead time of 0 is assumed. This means that the spare parts are immediately ready and no waiting time is required until the spare part is delivered. The data sets are not continuous in time. They have discrete timestamps.

4.1 Industrial data sets

A table summarizing the description of the data sets can be found below.

The first data set includes sales of 1392 *(3451 before cleaning)* [26] items of a dutch manufacturing company and will be named "MAN". The collection of the data started in the first week of 2012 until the 16th November of 2014 (150 weeks). It includes variables like prices, inventory

[26] Cleaning consists of dropping items which do not have > 1 demand occurrence in the train and test set, which are required for a forecast.

costs, the lead time , demand frequency and demand size, the minimum order quantity, the fixed order costs and the demand dates (per week).

The second industrial data set is gathered over seven years (1996-2002) and contains information about the demand of 5000 aircraft spare parts of the British Royal Air Force. The variables of this data set are nearly identical to the ones for the first data set, except for the inventory costs, which are not included in the "BRAF" data.

The third data set contains data from the "OIL" industry. It contains data about 7644 *(14523 before cleaning)* spare parts of an oil refinery for a period of 56 months (January 1997 to August 2001). This data set includes the prices and lead times.

Last but not least, a data set from the automotive industry. It contains, for instance, sales of 3000 items during 2 years. Again, the included variables are identical to the previous data sets, except that it does not contain price or lead time information. This is why the provided prices for "AUTO" in Table 6, have been calculated, by examining the relationship between pricing and monthly order frequency in the other data sets. Haan (2021) provides the formula for the ratio RPS (Ratio Price Sales), which allows to examine the relationship, but also the way to calculate the other price statistics of the Auto data set.

$$RPS = \frac{\text{Average item price}}{\text{Average monthly item sales}}$$

The RPS of the AUTO data set is set as the average of the RPS of the other data sets, which can be found in Table 7. The average product price is obtained by multiplying the RPS by the monthly sales. Haan (2021) and Nguyen (2023) also calculate the RMS (Ratio Monthly Sales), which is used to obtain the individual item price. The RMS is calculated as follows:

$$RMS = \frac{\text{Average monthly individual item sales}}{\text{Average monthly item sales}}$$

The Individual item price is calculated as follows:

$$\text{Individual item price} = \frac{\text{Average item price}}{\text{RMS}}$$

As Haan (2021) correctly points out, the mean product price for the AUTO data set is very high compared to the other data sets. This is due to the fact that the average product price is affected by the higher frequency portions, that can be found in the data set next to the relatively low average monthly item sales. A small, negative correlation is observed between the monthly average demand (item sales) and the average product price for the MAN, BRAF and OIL data sets. The AUTO data set shows a higher negative correlation. This is due to the fact that the prices of the AUTO data set have been calculated by respecting the ratios (RPS & RMS) of the other data sets. All the correlations are of a significance level of at least 5%.

Data set	Nr. Sales	Duration	Prices	Inventory costs	Lead time	Demand frequency & size	Min. order quantity	Fixed order costs
MAN	1392 items	150 weeks	Yes	Yes	Yes	Yes	Yes	Yes
BRAF	5000 items	7 years	Yes	No	Yes	Yes	Yes	Yes
Automotive industry	3000 items	2 years	No	Yes	No	Yes	Yes	Yes
OIL	7644 items	56 months	Yes	No	Yes	No	No	Yes

Table 6: This table summarizes the description of the industrial data sets.

Data	Monthly item sales				Product price				RPS	Corr. coeff.
	min	mean	max	SD	min	mean	max	SD		
MAN	0	24.22	4599.65	139.29	€ 0.09	€ 19.96	€ 297.54	€ 31.36	0.824	-0.0839**
BRAF	0.04	1.44	65.08	3.62	£ 0.001	£ 102.32	£ 9131.99	£ 373.33	70.943	-0.0885**
AUTO	0.54	4.45	129.17	7.57	€ 32.60*	€ 946.18*	€ 7772.86*	€ 1369.32*	212.633*	-0.4777**
OIL	0.04	0.63	232.73	4.02	€ 0.01	€ 355.85	€ 20493.17	€ 1076.12	566.132	-0.0417**
*Added by using the RPS and RMS calculations described above. (213.633 = (0.824 + 70.943 + 566.132) / 3).										
**p-value < 5%										

Table 7: Descriptive statistics for the MAN, BRAF, AUTO and OIL data sets

4.2 Simulated data sets

Simulated data sets allow to replicate a certain behaviour. In our case, every simulated data set replicates one of the four data categories (Erratic, Lumpy, Smooth, Intermittent). By having a clear dominating class of items, it is easier to conclude which method performs best for which data set. This facilitates the control of the environment and whether they have an impact on the performance. The four simulated data sets are generated in R, through the R package *'tsintermittent'* by Kourentzes (2014). Furthermore, we rely on Haan (2021)'s procedure. The package requires three input arguments. The three input arguments are:

1. Number of time series (1 per item)
2. Number of observations per time series
3. CV^2 and the average interval of the non-zero demand p

To resemble the industrial data sets, Haan (2021) sets the number of time series to 60 months, and the number of observations per time series to 6500 items. To be able to replicate every data category, the squared coefficient of variation, CV^2, and the mean inter-demand interval of non-zero demand, p, need to be chosen for every data set. This is done according to the cut-off values set by Boylan et al. (2008) in Section 3.4. Table 8 shows the settings for the four simulated data sets. Same as Haan (2021), we observe that average monthly demand decreases when p increases. The average product price is determined by using the average RPS of 212.633 by following the process in Section 4.1. A negative, significant correlation between monthly average demand and average product price is observed in all simulated data sets. This means, that the items of high demand are also cheaper items.

Furthermore, it is noticeable, that the negative correlation coefficient is much stronger for the simulated data sets (See Table 8) than for the industrial data sets (See Table 7. The industrial data sets include more non-zero demand occurrences than the simulated data sets. This can also be seen, when comparing the mean inter-demand interval, p in Table 9. In other words, the intermittency effect is much stronger in the industrial data sets, as there are nearly no zero demand occurrences in the simulated data sets. This raises the question if the simulated data sets are really replicating the behaviour of the industrial data sets, as the simulated data sets do not take into account the price as an input during the simulation process.

Data	Intended demand pattern	CV^2	p	Monthly demand		Product price*		Corr. coeff.
				mean	SD	mean	SD	
SIM1	Erratic	0.75	1.00	10.01	1.12	€ 2129.30	€ 246.06	-0.9872***
SIM2	Lumpy	0.80	1.50	6.66	1.12	€ 1416.52	€ 254.80	-0.9706***
SIM3	Smooth	0.30	1.05	9.50	0.74	€ 2019.21	€ 159.06	-0.9938***
SIM4	Intermittent	0.25	1.45	6.90	0.81	€ 1466.48	€ 180.19	-0.9848***

*Added by using the RPS and RMS calculations described above. ***p-value = 2.2e-16

Table 8: Settings for the simulated data sets

4.3 Classification of the data sets

As previously mentioned, the industrial and simulated data sets need to be classified. For this, the classification scheme of Boylan et al. (2008) is used. The scheme provides the important cut-off values of $p = 1.32$ and $CV^2 = 0.49$. The formulas that are used to calculate CV^2 and p for every individual item can be found in Section 3.4.

The results from the classification of Nguyen (2023) can be seen in Table 9. We observe that the inter-demand interval p of the industrial data sets is much higher than for the simulated data sets, except for the AUTO data set. The AUTO data set is the only data set, that cannot be classified as a single category, as it seems to have items for every demand type. However, the majority are classified as smooth and intermittent. The low inter-demand interval can be explained due to the high number of smooth items. Smooth items have frequent demand with low demand size variability (Boylan et al., 2008). The same is observed for SIM3, where nearly all the items are classified as being smooth items. SIM3 also shows the lowest inter-demand interval. Regarding the simulated data sets, we observe that they have been correctly classified. Nonetheless, we also observe that the simulated data sets do not only consist of purely one type of demand. SIM1 has mostly erratic items, SIM2 mostly lumpy items and SIM4 mostly intermittent items. With this classification, we will be able to answer which methods perform best on what kind of demand respectively for which data set.

Data	CV^2	p	Erratic items	Lumpy items	Smooth items	Intermittent items	Total items
MAN	0.92	16.41	23	806	1	562	1392
BRAF	0.63	11.14	0	2095	0	2905	5000
AUTO	0.41	1.32	378	307	1241	1074	3000
OIL	0.18	14.52	0	814	0	6830	7644
SIM1	0.75	1.00	6198	0	302	0	6500
SIM2	0.80	1.50	410	5614	25	451	6500
SIM3	0.30	1.05	36	0	6464	0	6500
SIM4	0.25	1.45	1	7	786	5706	6500

Table 9: Classification of the data sets by Nguyen (2023)

5 Results and analysis

In this section, we first compare the performance of each method for the different data sets and forecasting accuracy measures. Then, in a second stage, the inventory performance is analysed and compared to the forecasting accuracy results.

Method	Measure	Data set								# Best
		SIM1	SIM2	SIM3	SIM4	MAN	BRAF	AUTO	OIL	
Croston	MSE	79.202	79.138	34.730	40.519	12940.104	**199.690**	86.344	138.605	1
	MASE	0.673	1.027	0.487	0.780	2.499	2.080	0.788	1.998	0
	RMSSE	2.722	3.346	1.878	2.412	5.329	3.300	1.721	1.666	0
	GMAE	4.605	4.725	3.138	3.793	6.481	1.669	2.422	0.852	0
SBA	MSE	78.623	78.834	**34.620**	40.435	**12921.667**	199.807	83.089	**132.321**	3
	MASE	0.664	1.012	0.484	0.778	2.439	2.001	0.777	1.849	0
	RMSSE	2.712	3.337	**1.874**	2.409	**5.304***	3.289	**1.710**	1.635	2.5*
	GMAE	4.503	**4.592**	**3.102**	3.792	6.236	1.534	**2.362**	0.781	3
DLP	MSE	89.679	85.074	44.229	45.850	13002.323	203.685	112.859	148.810	0
	MASE	0.748	1.104	0.559	0.824	2.589	2.218	0.938	2.212	0
	RMSSE	2.918	3.488	2.126	2.562	5.372	3.362	2.031	1.740	0
	GMAE	5.305	5.227	3.671	3.970	6.903	1.872	3.036	0.987	0
MLP	MSE	78.087	**77.406**	34.842	**39.718**	13227.733	201.303	**82.672**	153.535	3
	MASE	0.679	1.027	0.493	**0.776**	3.121	2.301	0.822	2.066	1
	RMSSE	**2.708**	**3.316**	1.881	**2.388**	5.494	3.355	1.736	1.674	3
	GMAE	4.729	4.763	3.209	**3.783**	6.214	2.087	2.686	0.988	1
LSTM	MSE	87.170	86.052	70.132	53.216	865142.462	929.221	108.998	137.520	0
	MASE	0.661	1.121	0.698	0.856	19.087	5.816	0.866	1.730	0
	RMSSE	2.791	3.436	2.651	2.720	30.689	6.064	1.820	**1.465**	1
	GMAE	**4.174**	5.459	4.591	4.330	112.417	9.369	2.648	1.067	1
LightGBM	MSE	81.914	81.795	35.957	41.700	13584.090	202.644	95.428	156.013	0
	MASE	0.692	1.053	0.498	0.790	2.974	2.336	0.857	2.077	0
	RMSSE	2.772	3.410	1.909	2.454	5.602	3.375	1.866	1.686	0
	GMAE	4.727	4.756	3.209	3.841	5.958	1.897	2.612	0.926	0
RF	MSE	81.042	82.816	35.965	42.761	13385.695	201.351	88.078	154.925	0
	MASE	0.699	1.073	0.501	0.800	2.942	2.332	0.838	2.072	0
	RMSSE	2.761	3.436	1.911	2.476	5.544	3.372	1.790	1.683	0
	GMAE	4.893	4.922	3.247	3.898	6.064	1.950	2.578	0.940	0
Willemain	MSE	**77.928**	78.606	34.880	40.755	13111.759	199.775	84.182	132.939	1
	MASE	0.690	1.045	0.497	0.783	2.594	2.344	0.899	2.319	0
	RMSSE	2.714	3.348	1.886	2.420	**5.304**	3.365	1.863	1.726	0.5*
	GMAE	4.909	4.904	3.273	3.786	**4.392**	**1.082**	2.648	**0.613**	3
Quantile reg.	MSE	83.241	87.917	34.757	40.800	14104.492	202.800	88.210	139.344	0
	MASE	**0.638**	**0.956**	**0.477**	0.779	**1.586**	**1.167**	**0.746**	**0.956**	7
	RMSSE	2.779	3.512	1.877	2.421	5.371	3.253	1.724	1.487	0
**	*GMAE*	*1.715*	*1.698*	*0.954*	*1.689*	*0.479*	*0.000*	*1.366*	*0.000*	***8*

Results rounded to three decimals. The best accuracy is highlighted for each data set and measure.

*0.5, because only 50% is accounted to the method, as the place is shared with another method.

** The GMAE results of the QR need to be analyzed with caution.

Table 10: Forecasting accuracy of all the methods on each data set.

5.1 Forecasting accuracy measures

The values of the forecasting accuracy measures of all the methods applied to the data sets can be found in Table 10 with the best accuracy score highlighted for each method. The column # *Best* shows how many times the method is the best performer for the given accuracy measure and across all the data sets. The row of the Quantile regression method showing the GMAE accuracy is in italics as these results need to be analyzed with caution.

When blindly comparing all the results with each other, the quantile regression method is the best performer by far. It outperforms every other method in terms of GMAE and MASE, except for the MASE metric applied to the SIM4 data set. Here, MLP is the superior method. However, the GMAE results of the Quantile regression method show 0 values twice. This is due to the sensitivity of GMAE towards 0 absolute errors. In fact, as described in Subsection 3.3 and visualized in Table 5, GMAE equals zero as soon as one single observation has an absolute error of 0. As can be seen in Table 10, Quantile regression shows zero error for the BRAF data set and the OIL data set in terms of GMAE. For the other data sets, the GMAE is also low compared to the GMAE computed by the other methods. The low GMAE is due to the prediction of the Quantile regression method. The method predicts a lot of zeros when in fact there is some demand. Due to these particular results for Quantile regression in terms of GMAE, we decided to not include the GMAE accuracy performance of the QR method to calculate the Percentage Better score.

The quantile regression method is the best performer for the given accuracy measures and across all the data sets. It outperforms every other method in terms of MASE, except for the MASE metric applied to the SIM4 data set. Here, MLP is the superior method. The worst performing methods are DLP, LightGBM and RF as, in our experiment, they fail to outperform all the other methods in one instance.

MLP shows the best performance for the simulated data sets, whereas SBA outperforms the other methods in most instances of the industrial data sets. In fact, the superiority is due to the fact that the GMAE accuracy measures are not included for the QR method in the comparison for the previously given reasons.

The second best performing method is SBA. It performs well on both data sets types, especially in terms of MSE and GMAE. MLP ranks 3rd. It is superior in terms of MSE and RMSSE. When focusing on the data sets, MLP outperforms all the other methods for the SIM4 data set in terms of MSE, MASE, RMSSE and GMAE. It outperforms every other method in 8 instances.

Following MLP, in fourth place, Willemain can be found. Willemain outperforms the other methods in 4.5* instances, i.e. MSE for SIM1 and shares the place with SBA for the RMSSE for MAN. Furthermore, in terms of GMAE, it is superior for the MAN, BRAF and OIL data sets. In fifth place is LSTM showing superiority in two instances, followed by Croston (superiority in one instance). It is noteworthy that LSTM has not been extensively tuned due to fear of overfitting. In fact, as the data becomes scarce when creating the sequences, there is less data available for tuning and testing. Hence, the model is kept simple.

The only methods, that do not outperform the other methods in one instance, are DLP, LightGBM and RF.

The Percentage Better score is computed by dividing the number of times a method outperforms every other method by the total number of times the method has been used. For example, Croston is the best performer once. We divide 1 by 32 (as we have 32 instances) *100 and obtain 3.125%. The same calculation is done for every method. Quantile regression is superior to the other methods in 29.167% of the comparisons (for MSE, MASE and RMSSE), followed by SBA with 26.563% and MLP with 25%.

Another observation that is made while running the methods is, that the ML methods take much longer time to run. Especially RF and LightGBM, nevertheless, they do not perform well in terms of forecasting accuracy. Another computing time consuming method is Willemain's method, which is due to the bootstrapping. The run-times of the methods have not been measured in RStudio, however, the duration has been observed.

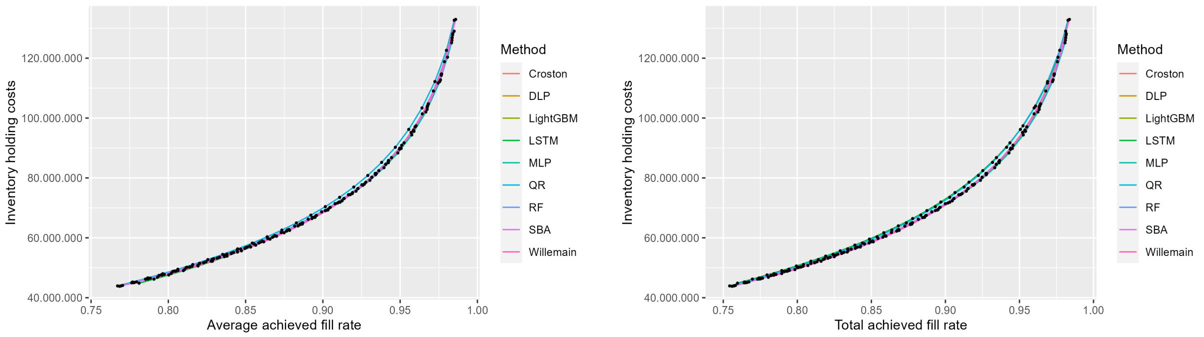
Croston	SBA	DLP	MLP	LSTM	LightGBM	RF	Willemain	Quantile regression
3.125%	26.563%	0%	25%	6.25%	0%	0%	14.063%	29.167%

Table 11: Percentage Better score of the methods.

5.2 Inventory performance

5.2.1 Inventory performance of the simulated data sets

Figure 3a and Figure 3b show the trade-off curves between the achieved fill rate (AFR) and the inventory holding costs for the SIM1 data set. Plot (a) shows the average achieved fill rate, whereas plot (b) shows the total achieved fill rate. The SIM1 data set consists mostly of erratic items and a small part of smooth items. When comparing both plots, the trade-off curves show a similar pattern. In Plot (a), the quantile regression (QR) method shows higher inventory holding costs for the same average AFR as the other methods. The other methods are bundled together and behave similarly. When looking at the total AFR vs inventory holding costs trade-off curves, LSTM and QR stand out from the other methods, as their inventory holding costs are higher for the same total AFR. In fact, up to 83% total AFR, they behave similarly to the other methods. From 83% to 97%, they decouple. In the Appendix 7.1, the table with all the values for the average AFR, total AFR and the inventory holding costs is provided.

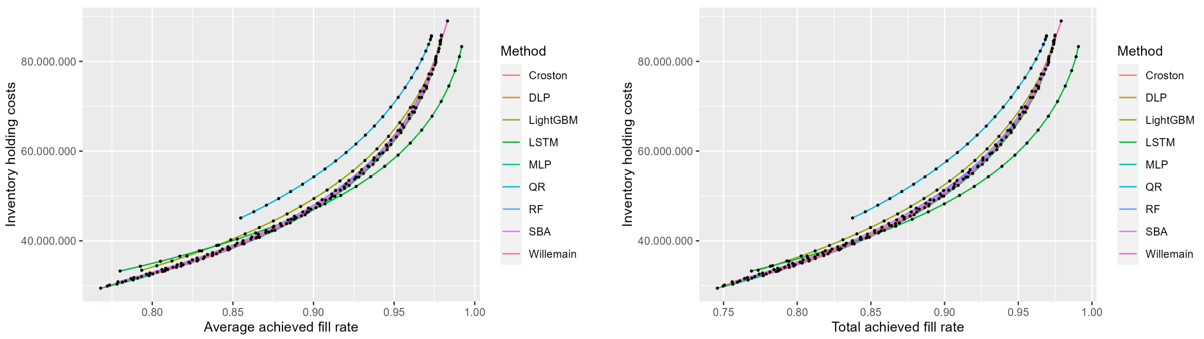


(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 3: Trade-off curves for the inventory control measures on SIM1

The inventory performance results of the SIM2 data set, which consists mostly of lumpy items, are shown in Figure 4. Both plots, 4a and 4b show that Quantile regression has higher costs for the same AFR compared to the other methods. This is the case for the average AFR and total AFR. One method that stands out for the higher AFR, is LSTM. LSTM performs slightly worse for 75% AFR (total and average) as the other methods except QR, but decouples from the bundle (from 90% average AFR onwards and from 86% total AFR onwards), and outperforms all the other methods in terms of costs and AFR.



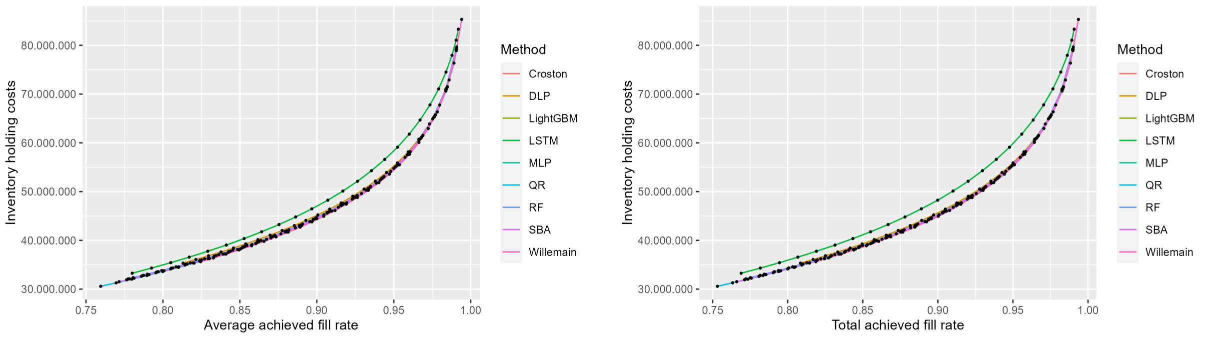
(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 4: Trade-off curves for the inventory control measures on SIM2

The results of the SIM3 data set, which is dominated by smooth demand, show that all the methods display a similar behaviour between AFR and Inventory holding costs, except for LSTM. LSTM consistently achieves the same fill rates (total and average) for a higher cost compared to the other methods.

The trade-off curves for the SIM4 data set can be found in the Appendix 7.1, as they do not provide more insights. Their behaviour is similar to the curves for SIM3.



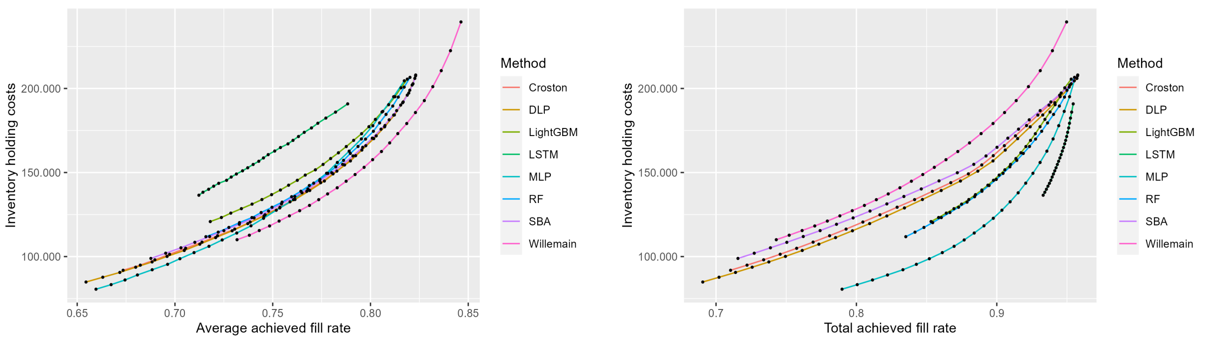
(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 5: Trade-off curves for the inventory control measures on SIM3

5.2.2 Inventory performance of the industrial data sets

The MAN data set is characterized by lumpy and intermittent items. Plot 6a shows that Willemain presents the lowest costs for the same AFR as the other methods. In general, Willemain also achieves higher fill rates than the other methods. LSTM seems to provide the highest costs for the range of 0.71% AFR to 0.77%. However, this is not the case for the total AFR. In plot 11b, Willemain is outperformed by all the other methods. Especially, the ML methods perform very well. QR has been removed from this plot, as due to its low performance in terms of AFR, it was twisting the plot. The plot including QR can be found in the Appendix 7.1 Figure 11.



(a) Average achieved fill rate vs Inventory holding costs

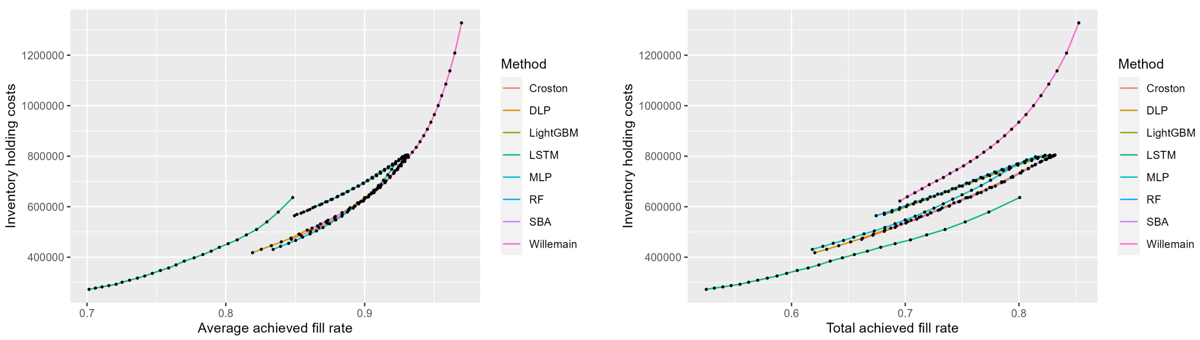
(b) Total achieved fill rate vs Inventory holding costs

Figure 6: Trade-off curves for the inventory control measures on MAN

The BRAF data set is characterized lumpy items and intermittent items. The average AFR plot 7a and the total AFR plot 7b differ. Plot 7a shows the highest costs for LSTM for the range of 70% to 85% average AFR. LSTM does not achieve higher average fill rates than that. Another method that stands out, is Willemain. Willemain provides the highest average AFR. Willemain starts standing out from the other methods from the average AFR of 93%, where the costs increase exponentially. The other methods do not stand out, except for MLP, which in the range of 83% average AFR to 87% provides the same average fill rate for lower costs. When comparing the total AFR to the Inventory holding costs, LSTM stands out for having the lowest costs. However, the total AFR caps at 80% for LSTM. Whereas all the other methods achieve higher fill rates. Especially Willemain achieves the highest total fill rates (85%) but for much higher costs.

In both plots, QR has been removed as it does not provide any insights at all. In fact, it does not show a curve at all, average AFR, total AFR and the inventory holding costs are 0. This is due to the predictions of the QR method for the BRAF data set and the test BRAF data set (actual demand). The prediction data frame consists of only zeros, which then has an impact on the fill rate and the holding costs. The fill rate is calculated as the total supply divided by the total demand, and the holding costs are proportional to the inventory level, which would be zero if there is no demand at all. Another argument could be that the test data contains mostly zero demand. This means that the achieved fill rates would also be zero since there is no demand to fulfil. The same logic for the inventory holding costs, which would also be zero since there is no need to hold inventory.

The same is observed for the OIL data set. Both, the OIL and BRAF data sets consist of lumpy and intermittent items only with lots of 0 demand values. In Appendix 7.1 Table 17 summarizes the behaviour of the QR method for the BRAF data set.

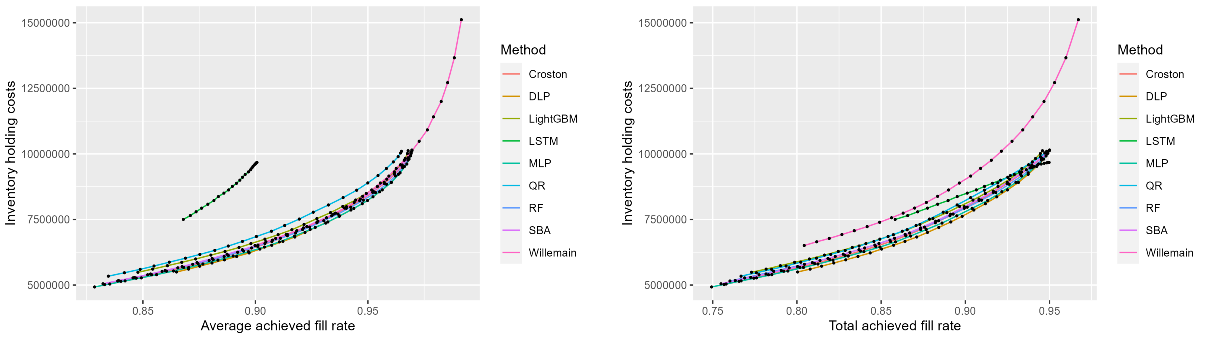


(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 7: Trade-off curves for the inventory control measures on BRAF

The AUTO data set consists of items out of the four categories, with the majority being smooth items and intermittent items as shown in Table 9. Willemain achieves the highest fill rates for the average AFR and total AFR. However, for those higher AFR, the costs are also higher. The other methods fail to achieve the same fill rates. When looking at the figures 8a and 8b separately, it is visible that for the former all the methods, except LSTM are clumped together up to the average AFR of 96%. Above those fill rates, only Willemain succeeds. Furthermore, LSTM has higher inventory holding costs, but also a lower average AFR (only up to 90%). When looking at the total achieved fill rate, Willemain is outperformed by every other method in terms of inventory holding costs. However, in terms of total AFR, Willemain achieves slightly higher fill rates (1.7% higher) compared to the other methods.

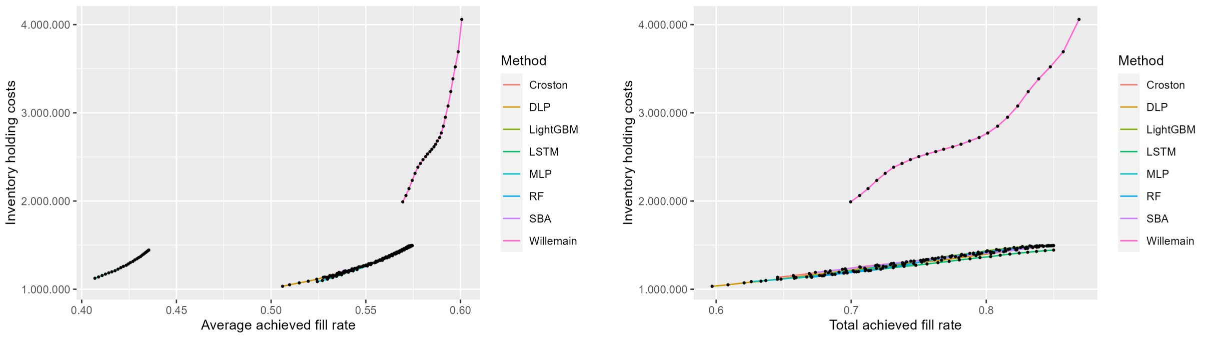


(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 8: Trade-off curves for the inventory control measures on AUTO

Last, but not least, the results of the inventory performance of the OIL data set are shown. The OIL data set is characterized by lumpy and intermittent items. Two aspects that jump out when looking at Figure 9, are the performance of LSTM in Figure 9a. LSTM performs badly in terms of average AFR compared to the other methods for the same inventory holding costs, except compared to Willemain. The second aspect that jumps out is, that Willemain achieves the highest average AFR at 60%. However, the costs are also much higher. The other methods are bundled together and perform similarly. Figure 9b shows the same behaviour as in Figure 9a, except that this time, LSTM does not stand out on its own. This time, LSTM performs similarly to the other methods. This difference in the behaviour of LSTM for the average AFR and total AFR can be attributed to the nature of the OIL data set. The OIL data set contains lots of 0 demand. Furthermore, the average interval of the non-zero demands p is high, meaning that the intermittency effect is much stronger in the OIL data set. Hence, the different behaviour on average for the highly intermittent data set, consisting of slow-moving items, which can be observed in Table 6 (average monthly item sales of 0.63 items).



(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 9: Trade-off curves for the inventory control measures on OIL

6 Discussion and Conclusion

In this section, first, the findings of this paper are discussed. Next, the findings are compared to other existing papers. Followed by linking the findings to the research questions. Finally, a conclusion and further possible research are brought up.

In this paper, nine different methods from three categories were compared. Namely, Croston, Syntetos-Boylan approximation (SBA) and DLP grouped into statistical methods. Machine learning methods consisting of Multi-Layer Perceptron (MLP), Long-Short term memory (LSTM), LightGBM and Random Forest (RF). The third category, Non-parametric methods is formed, by Willemain's bootstrapping method and Quantile regression. The nine methods were applied to eight different data sets. Four industrial and four simulated data sets, simulate a certain demand behaviour. To measure the performance of the methods, forecasting accuracy measures (MSE, MASE, RMSSE and GMAE) and inventory control measures (Achieved fill rate and Inventory holding costs) were used.

6.1 Findings

Finding 1

Differences in results based on the performance measure used and data set category.

Throughout this paper, we have demonstrated, that the performance of the method depends on the performance measures used. In fact, the best performing methods in terms of forecasting accuracy are not necessarily the best performing methods in terms of inventory performance. The Percentage Better comparison has shown, that Quantile regression outperformed the other methods followed by SBA. DLP, LightGBM and RF were overall the worst methods. Based on Inventory control measures, Willemain outperforms the other methods. However, for lumpy demand LSTM outperforms Willemain. MLP stands out as the best performer for erratic demand. DLP outperforms the other methods for the AUTO data set in terms of total AFR and inventory holding costs.

Finding 2

Data cleaning plays a crucial role.

The handling of the data is very important. The results can differ because of the different data cleaning. For example, Pennings et al. (2017) cleans the BRAF data differently compared to us. In their paper, the BRAF data set consists of only 1131 SKUs, whereas in our paper, it consists of 5000 SKUs. No explanation was given on how the data was cleaned by Pennings et al. (2017). This results in different accuracy measures for the same methods. For Croston and SBA, Pennings et al. (2017) reports better results in terms of MASE compared to us. However, the opposite is noted for GMAE. Furthermore, the DLP method performs better in both instances (MASE and GMAE) compared to Pennings et al. (2017) for the BRAF data set.

Finding 3

The cost of hyper parameter tuning of the ML methods.

Machine learning methods are represented in the top performer rankings of both performance measures, i.e. for forecasting accuracy and inventory performance. One major point of ML methods is the tuning of the hyper parameters, which can improve the performance of the model. For example, by adding a hyper parameter tuning grid and trying multiple combinations of values for the parameters, we obtained better results for LightGBM for all data sets and all metrics compared to Haan (2021), who does not try multiple values.

However, the problem with hyper parameters tuning is that it is time-consuming and requires knowledge about the different parameters and how their values have an on the model, which is an important point for the reproducibility of the methods.

Finding 4

GMAE sensitivity for values of 0.

As observed in Table 10, in two instances the GMAE metric equals 0. This is the case for the BRAF and OIL data set when the QR method is applied. As previously explained, the QR method predicts many 0 demands for many periods, items and quantiles. In combination with

0 actual demand, this causes the absolute errors to equal 0. Due to these 0 absolute errors, the geometric mean becomes automatically 0 even if other absolute errors are > 0. This is only observed for the quantile regression method, as it is the only method that predicts 0 demand when there actually is 0 demand. Therefore, the sensitivity of GMAE for 0 values is a finding that should be always considered in combination with the used method.

6.2 Comparison of the findings with the reviewed literature

Haan (2021) review

In the paper of Haan (2021) [27]., seven methods are run across eight data sets. Namely, Croston, SES, SBA, TSB, Willemain, MLP and LightGBM. Out of those seven methods, we also run Croston, SBA, Willemain, MLP and LightGBM. In fact, our paper is an extension of Haan (2021)'s paper as it takes some of their methods plus new methods (LSTM, RF, DLP and Quantile regression) and runs them on the same eight data sets. The performance measures are the same. In both papers, MSE, MASE and RMSSE are used as forecasting accuracy measures and the trade-off curves of service levels are used as inventory performance measures. Our paper goes one step further and adds a new forecasting accuracy measure, namely GMAE. In Haan (2021)'s study, SBA is superior overall based on the Percentage Better comparison. SBA performs second best after Quantile regression based on the Percentage Better score in our paper. This shows that in both papers SBA proves itself as a reliable method. In terms of inventory performance, our paper draws the same conclusion as Haan (2021). Willemain's bootstrapping method is overall superior. Regarding the ML methods, LightGBM is outperformed by every method in both papers, although tuned differently, in terms of forecasting accuracy. MLP is the second best performing method in Haan (2021) for forecasting accuracy and the third best, in this paper. This proves that Haan (2021)'s findings are reproducible.

Theodorou et al. (2023) review

In the recent, yet not published paper of Theodorou et al. (2023), they conduct a comparison study with eleven methods. Three methods are also used in our paper, namely, Croston, SBA and LightGBM. These methods are applied to one single data set (retail sales from Makridakis et al. (2022)). The data set consists of mostly intermittent (73%) and lumpy (17 %) items (3% erratic and 7% smooth). In our paper, the OIL data set, BRAF data set and SIM4 show similar demand characteristics. LightGBM stands out for both RMSSE and trade-off curves for the

[27] A newer version of this paper is the working paper in progress, submitted to the International Journal of Production Economics; October 2, 2023

higher review periods. Croston and SBA do not show significant superiority compared to the other methods. The performance of LightGBM in this paper complies with that in ours. In fact, we do not analyze the behaviour for higher review periods. However, the low performance of LightGBM in general is in accordance with our findings.

6.3 Conclusion

In this section, the findings are linked to the research questions of this paper. The first question is as follows: *"Which methods perform best on what kind of demand respectively for which data set?"*

Throughout this paper, the performance of 9 different spare parts demand forecasting methods from 3 different categories have been studied. The methods are categorized as *Statistical methods, Machine learning methods and Non-parametric methods.* 8 data sets characterized by certain demand behaviours are used to run the methods. The demand is classified as either *lumpy, erratic, intermittent or smooth.* The findings suggest that there is no consistent superior method based on the data set. SBA shows superiority in 3 out of 4 accuracy metrics for the SIM3 data set (Smooth demand), and 2 out of 4 accuracy metrics for the AUTO data set. However, these findings are not consistent in terms of inventory performance measures. For the same data sets, different methods are superior. The SIM3 data set shows the best results for Willemain's method. Whereas the OIL data set provided the best results for the MLP method in terms of average achieved fill rate and LSTM in terms of total achieved fill rate. SIM4 on the other hand shows superiority in all four accuracy metrics when the MLP method is applied to it. Again, this is not consistent with the findings based on the inventory performance measure. Willemain is superior here. Therefore, no definitive answer can be given to this research question as consistency is lacking.

The second question of this paper is as follows: *"Is the performance of certain methods due to the measure used?"*

This question is aimed at the use of forecasting accuracy measures and how they impact prediction accuracy. Table 10 shows indeed that some accuracy measures perform consistently well for a method throughout most data sets. The most eye-catching example is the performance of the quantile regression (QR) method for the MASE accuracy measure. In fact, the MASE accuracy measure for quantile regression provides the lowest MASE in 7 out of 8 instances. Another noteworthy example is GMAE, which provides superior performance for Willemain on 3 out of 4 industrial data sets. Or the RMSSE performance for MLP on 3 out of 4 simulated data sets. The reason for this is out of this paper's scope. However, future research can be conducted on this.

The third and final research question of this paper is: *"Do machine learning methods perform better in general than statistical methods?"*

Machine learning methods are widely used in other domains as they seem to perform well. In the spare parts demand forecasting field, machine learning methods are not widely used as their performance is not yet fully studied. Furthermore, some ML methods are not easily understandable as they include a black box problem. In this paper, a total of four ML methods have been used. Namely, Multi-layer perceptron (MLP), Long-short term memory (LSTM), LightGBM and Random forest (RF). Furthermore, 3 out of the 9 methods are categorized as statistical methods in Section 3.1. When looking at Table 10, the statistical methods are superior in 9.5 out of 96 instances, whereas the ML methods are superior in 10 out of 128 instances. Furthermore, LightGBM and RF, both fail to show superiority in at least one instance. For the statistical methods, DLP does not perform best for one accuracy metric. This shows that in terms of forecasting accuracy, the statistical methods perform better on average than the ML methods.

However, when looking at the inventory performance measure, the opposite is observed. In fact, only DLP from the statistical methods achieves the highest fill rate for the total AFR. ML methods on the other hand, are more often the best performing method when it comes to inventory performance. Consequently, no general conclusion can be drawn from our paper. Although ML methods provide promising results in terms of inventory performance, they lag behind statistical methods in terms of forecasting accuracy.

6.4 Discussion

For future research, ML methods in spare parts demand forecasting should further be studied as there are many aspects of the implementation that can be analysed. One is the training of the model. Do models, that have been trained through single SKU [28] training perform better than cross SKU trained models. Furthermore, the hyper parameter tuning of the ML methods plays a big role. One could pay more attention to the behaviour of the hyper parameters in the spare parts demand forecasting domain, to be able to determine which methods are easy to implement, i.e. not complicated to build and tune, do not require much computational power and work with little data.

[28] Stock keeping unit

6.5 Acknowledgments

I would like to express my deepest gratitude to Dekker for the invaluable guidance throughout this thesis. Furthermore, I would like to thank Dekker, Nguyen, De Haan, Syntetos, Boylan, Teunter, Duncan and Porras for providing the data sets. Lastly, I would like to mention my family and friends for supporting me through this process.

6.6 Data and programming code

The methods have been implemented in RStudio and on Google Colab. The code and the data sets can be found on GitHub. The URL for the GitHub repository is https://github.com/YllorH/SpareParts_MasterThesis.

7 References

Aktepe, A., Yanık, E., & Ersöz, S. (2021). Demand forecasting application with regression and artificial intelligence methods in a construction machinery company. *Journal of Intelligent Manufacturing*, *32*, 1587–1604.

Arya Poddar, K. D., Aiana Goyal. (2019). Scorecardmodelutils: Credit scorecard modelling utils. https://CRAN.R-project.org/package=scorecardModelUtils

Baryannis, G., Validi, S., Dani, S., & Antoniou, G. (2019). Supply chain risk management and artificial intelligence: State of the art and future research directions. *International Journal of Production Research*, *57*(7), 2179–2202.

Biau, G., & Scornet, E. (2016). A random forest guided tour. *Test*, *25*, 197–227.

Boylan, J. E., Syntetos, A. A., et al. (2006). Accuracy and accuracy-implication metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting*, *4*, 39–42.

Boylan, J. E., Syntetos, A. A., & Karakostas, G. C. (2008). Classification for forecasting and stock control: A case study. *Journal of the operational research society*, *59*, 473–481.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5–32.

Callioni, G., de Montgros, X., Slagmulder, R., Van Wassenhove, L. N., & Wright, L. (2005). Inventory-driven costs. *harvard business review*, *83*(3), 135–41.

Chandriah, K. K., & Naraganahalli, R. V. (2021). Rnn/lstm with modified adam optimizer in deep learning approach for automobile spare parts demand forecasting. *Multimedia Tools and Applications*, *80*(17), 26145–26159.

Choi, B., & Suh, J. H. (2020). Forecasting spare parts demand of military aircraft: Comparisons of data mining techniques and managerial features from the case of south korea. *Sustainability*, *12*(15), 6045.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society*, *23*(3), 289–303.

de Haan. (2021). GitHub repository for benchmarking spare parts demand forecasting for intermittent demand. https://github.com/danieldehaan96/spdf

Dennis, M. J., & Kambil, A. (2003). Service management: Building profits after the sale: After-sale services and parts can contribute as much as 50 percent of all profits for the typical manufacturing company. but to capitalize on that opportunity, you need a supply chain explicitly designed to support superior service management–and do so in a profitable way. building that ”service-to-profit” supply chain calls for a sage strategy and flawless execution. [42]. *Supply Chain Management Review*, *7*(1), 42(7).

Durlinger, P., & Paul, I. (2015). Inventory and holding costs. *Durlinger Consultancy, Posterholt, The Netherlands, Tech. Rep. WP, 4*.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26. Retrieved June 14, 2023, from http://www.jstor.org/stable/2958830

Gutierrez, R. S., Solis, A. O., & Mukhopadhyay, S. (2008). Lumpy demand forecasting using neural networks. *International journal of production economics*, *111*(2), 409–420.

Haan, d. (2021). *Benchmarking spare parts demand forecasting methods* (Master's thesis). http://hdl.handle.net/2105/60771

Haneveld, W. K., & Teunter, R. (1997). Optimal provisioning strategies for slow moving spare parts with small lead times. *Journal of the Operational Research Society*, *48*(2), 184–194.

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

Hyndman, R. J., & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, *22*(4), 679–688.

İfraz, M., Aktepe, A., Ersöz, S., & Çetinyokuş, T. (2023). Demand forecasting of spare parts with regression and machine learning methods: Application in a bus fleet. *Journal of Engineering Research*, *11*(2), 100057.

Kailex. (2020). M5 forecasting competition. https://www.kaggle.com/kailex/m5-forecaster-v2

keras-team. (2021). GitHub repository of the keras package/optimizers/adam.py. https://github.com/keras-team/keras/blob/v2.13.1/keras/optimizers/adam.py

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, *15*(4), 143–156.

Kolassa, S. (2016). Evaluating predictive count data distributions in retail sales forecasting. *International Journal of Forecasting*, *32*(3), 788–803.

Kourentzes, N. (2013). Intermittent demand forecasts with neural networks. *International Journal of Production Economics*, *143*(1), 198–206.

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

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, *281*(3), 628–641.

Learned-Miller, E. G. (2014). Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, 3.

Liaw, A., Wiener, M., et al. (2002). Classification and regression by randomforest. *R news*, *2*(3), 18–22.

Lolli, F., Gamberini, R., Regattieri, A., Balugani, E., Gatos, T., & Gucci, S. (2017). Single-hidden layer neural networks for forecasting intermittent demand. *International Journal of Production Economics*, *183*, 116–128.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PloS one*, *13*(3), e0194889.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2022). The m5 competition: Background, organization, and implementation. *International Journal of Forecasting*, *38*(4), 1325–1336.

Microsoft. (2021). *Welcome to lightgbm's documentation! - lightgbm 3.2.1.99 documentation.* https://lightgbm.readthedocs.io/en/latest/

Nguyen. (2023). GitHub repository for spare parts demand forecasting. https://github.com/KhueNguyenTK/Spare-Part-Demand-Forecasting

Pennings, C. L., Van Dalen, J., & van der Laan, E. A. (2017). Exploiting elapsed time for managing intermittent demand for spare parts. *European Journal of Operational Research*, *258*(3), 958–969.

Pinçe, Ç., Turrini, L., & Meissner, J. (2021). Intermittent demand forecasting for spare parts: A critical review. *Omega*, *105*, 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.

Probst, P., Wright, M. N., & Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, *9*(3), e1301.

Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, *1*(5), 206–215.

SciKit-Learn. (2015). Lstm model. https://github.com/scikit-learn/scikit-learn/blob/7f9bad99d6e0a3e8ddf92a7e5561245224dab102/sklearn/model_selection/_search.py#L1056

SciKit-Learn. (2019a). Forest of trees-based ensemble methods. https://github.com/scikit-learn/scikit-learn/blob/7f9bad99d6e0a3e8ddf92a7e5561245224dab102/sklearn/ensemble/_forest.py#L1090

SciKit-Learn. (2019b). Multilayer perceptron. https://github.com/scikit-learn/scikit-learn/blob/7f9bad99d/sklearn/neural_network/_multilayer_perceptron.py#L765

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Shi, Y., Ke, G., Soukhavong, D., Lamb, J., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., et al. (2022). Lightgbm: Light gradient boosting machine. r package version 3.3. 4.

Silver, E. A. (1981). Operations research in inventory management: A review and critique. *Operations Research*, *29*(4), 628–645.

Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, *36*(1), 75–85.

Spiliotis, E., Makridakis, S., Semenoglou, A.-A., & Assimakopoulos, V. (2020). Comparison of statistical and machine learning methods for daily sku demand forecasting. *Operational Research*, 1–25.

Suomala, P., Sievänen, M., & Paranko, J. (2002). The effects of customization on spare part business: A case study in the metal industry. *International Journal of Production Economics*, *79*(1), 57–66.

Syntetos, A. A., Babai, M. Z., & Gardner Jr, E. S. (2015). Forecasting intermittent inventory demands: Simple parametric methods vs. bootstrapping. *Journal of Business Research*, *68*(8), 1746–1752.

Syntetos, A. A., Babai, M. Z., & Altay, N. (2012). On the demand distributions of spare parts. *International Journal of Production Research*, *50*(8), 2101–2117.

Syntetos, A. A., & Boylan, J. E. (2001). On the bias of intermittent demand estimates. *International journal of production economics*, *71*(1-3), 457–466.

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

Syntetos, A. A., & Boylan, J. E. (2006). On the stock control performance of intermittent demand estimators. *International Journal of Production Economics*, *103*(1), 36–47.

Syntetos, A. A., Nikolopoulos, K., & Boylan, J. E. (2010). Judging the judges through accuracy-implication metrics: The case of inventory forecasting. *International Journal of Forecasting*, *26*(1), 134–143.

Taylor, J. W. (2007). Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational Research*, *178*(1), 154–167. https://doi.org/https://doi.org/10.1016/j.ejor.2006.02.006

Teunter, R. H., & Duncan, L. (2009). Forecasting intermittent demand: A comparative study. *Journal of the Operational Research Society*, *60*(3), 321–329.

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.

Theodorou, E., Spiliotis, E., & Assimakopoulos, V. (2023). Forecasting accuracy and inventory performance: Fallacies and facts. https://doi.org/10.13140/RG.2.2.26143.33446

Trapero, J. R., Cardós, M., & Kourentzes, N. (2019). Quantile forecast optimal combination to enhance safety stock estimation. *International Journal of Forecasting*, *35*(1), 239–250.

Van Jaarsveld, W., & Dekker, R. (2011). Estimating obsolescence risk from demand data to enhance inventory control—a case study. *International Journal of Production Economics*, *133*(1), 423–431.

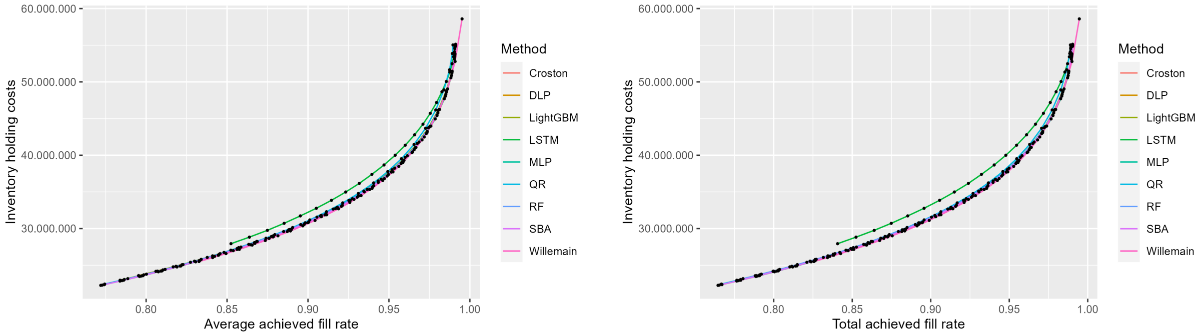
Van Wingerden, E., Basten, R., Dekker, R., & Rustenburg, W. (2014). More grip on inventory control through improved forecasting: A comparative study at three companies. *International journal of production economics*, *157*, 220–237.

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

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks:: The state of the art. *International journal of forecasting*, *14*(1), 35–62.

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.

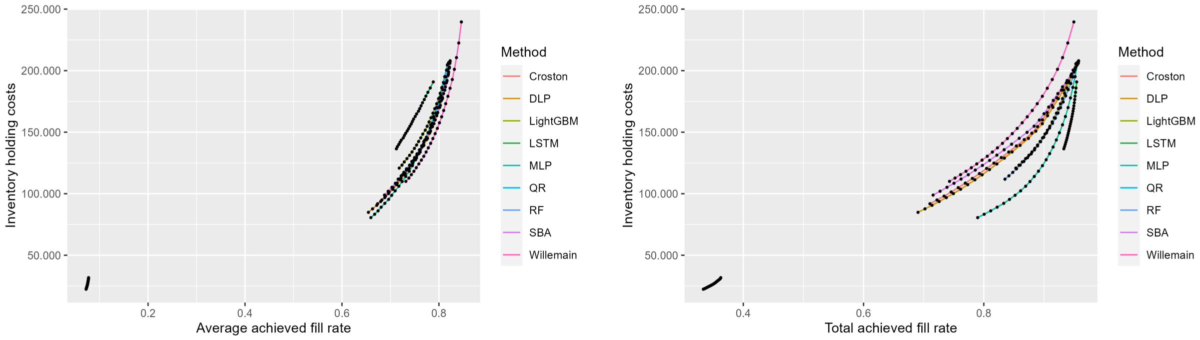
7.1 Appendix



(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 10: Trade-off curves for the inventory control measures on SIM4



(a) Average achieved fill rate vs Inventory holding costs

(b) Total achieved fill rate vs Inventory holding costs

Figure 11: Trade-off curves for the inventory control measures on MAN

Table 12: IPM values for SIM1

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.7704023	0.7582062	44089758	0.75	Croston
0.76	0.7793947	0.7673392	45274896	0.76	Croston
0.77	0.7884249	0.7765073	46503867	0.77	Croston
0.78	0.7973634	0.7856070	47774748	0.78	Croston
0.79	0.8062552	0.7946872	49115174	0.79	Croston
0.8	0.8152954	0.8039277	50512524	0.80	Croston
0.81	0.8243103	0.8131350	51982631	0.81	Croston
0.82	0.8331705	0.8222331	53525921	0.82	Croston
0.83	0.8422245	0.8315214	55153978	0.83	Croston
0.84	0.8510582	0.8406075	56882390	0.84	Croston

0.85	0.8601456	0.8499658	58717018	0.85	Croston
0.86	0.8690436	0.8591961	60673700	0.86	Croston
0.87	0.8780400	0.8684938	62774485	0.87	Croston
0.88	0.8869076	0.8777531	65034573	0.88	Croston
0.89	0.8959089	0.8871233	67490448	0.89	Croston
0.9	0.9047939	0.8964278	70176271	0.90	Croston
0.91	0.9137951	0.9058731	73138289	0.91	Croston
0.92	0.9227515	0.9153346	76448445	0.92	Croston
0.93	0.9317090	0.9248242	80187117	0.93	Croston
0.94	0.9406200	0.9343156	84492457	0.94	Croston
0.95	0.9495341	0.9438683	89568942	0.95	Croston
0.96	0.9585362	0.9535636	95769427	0.96	Croston
0.97	0.9675876	0.9633944	103681310	0.97	Croston
0.98	0.9766502	0.9733371	114274391	0.98	Croston
0.99	0.9839135	0.9814686	127721141	0.99	Croston
0.751	0.7685158	0.7559839	43815499	0.75	SBA
0.761	0.7776036	0.7652483	45016455	0.76	SBA
0.771	0.7867333	0.7745349	46249440	0.77	SBA
0.781	0.7957577	0.7837106	47526081	0.78	SBA
0.791	0.8047352	0.7928983	48875023	0.79	SBA
0.810	0.8139508	0.8023410	50283278	0.80	SBA
0.811	0.8229913	0.8116029	51758591	0.81	SBA
0.821	0.8320161	0.8208869	53317173	0.82	SBA
0.831	0.8411305	0.8302529	54953671	0.83	SBA
0.841	0.8500691	0.8394644	56699636	0.84	SBA
0.851	0.8592817	0.8489626	58545823	0.85	SBA
0.861	0.8682740	0.8583097	60523307	0.86	SBA
0.871	0.8773621	0.8677166	62635120	0.87	SBA
0.881	0.8863351	0.8770894	64923736	0.88	SBA
0.891	0.8954767	0.8866166	67400304	0.89	SBA
0.910	0.9044731	0.8960567	70118141	0.90	SBA
0.911	0.9135439	0.9055950	73104393	0.91	SBA
0.921	0.9225841	0.9151495	76447867	0.92	SBA

0.931	0.9316229	0.9247355	80233527	0.93	SBA
0.941	0.9406152	0.9343267	84585923	0.94	SBA
0.951	0.9496027	0.9439605	89726450	0.95	SBA
0.961	0.9586340	0.9536983	96004812	0.96	SBA
0.971	0.9677646	0.9636188	104018014	0.97	SBA
0.981	0.9768570	0.9735955	114727117	0.98	SBA
0.991	0.9840319	0.9816119	128165348	0.99	SBA
0.752	0.7873648	0.7759118	46537876	0.75	DLP
0.762	0.7955550	0.7842353	47692393	0.76	DLP
0.772	0.8037482	0.7925682	48893333	0.77	DLP
0.782	0.8120004	0.8009701	50148510	0.78	DLP
0.792	0.8200476	0.8091981	51438621	0.79	DLP
0.812	0.8282985	0.8176316	52806185	0.80	DLP
0.813	0.8364487	0.8259841	54227484	0.81	DLP
0.822	0.8445130	0.8342478	55719342	0.82	DLP
0.832	0.8526930	0.8426626	57297410	0.83	DLP
0.842	0.8607545	0.8509562	58959353	0.84	DLP
0.852	0.8689410	0.8594042	60718886	0.85	DLP
0.862	0.8769697	0.8677226	62594719	0.86	DLP
0.872	0.8852444	0.8762841	64607311	0.87	DLP
0.882	0.8932913	0.8846690	66759923	0.88	DLP
0.892	0.9015035	0.8932237	69101715	0.89	DLP
0.912	0.9095615	0.9016666	71641418	0.90	DLP
0.913	0.9177436	0.9102451	74441608	0.91	DLP
0.922	0.9259539	0.9189073	77562812	0.92	DLP
0.932	0.9341353	0.9275728	81070764	0.93	DLP
0.942	0.9423636	0.9363263	85092823	0.94	DLP
0.952	0.9506704	0.9452025	89831087	0.95	DLP
0.962	0.9589796	0.9541317	95574033	0.96	DLP
0.972	0.9674741	0.9633364	102885846	0.97	DLP
0.982	0.9761112	0.9727749	112757190	0.98	DLP
0.992	0.9835301	0.9810395	125993925	0.99	DLP
0.753	0.7929721	0.7804459	46818397	0.75	Willemain

0.763	0.8017884	0.7894467	48078462	0.76	Willemain
0.773	0.8104588	0.7983648	49388375	0.77	Willemain
0.783	0.8190186	0.8071413	50768368	0.78	Willemain
0.793	0.8276664	0.8160500	52189948	0.79	Willemain
0.814	0.8362949	0.8249322	53696521	0.80	Willemain
0.815	0.8447889	0.8337232	55264826	0.81	Willemain
0.823	0.8532402	0.8424672	56923391	0.82	Willemain
0.833	0.8617314	0.8512812	58667771	0.83	Willemain
0.843	0.8700370	0.8599152	60523158	0.84	Willemain
0.853	0.8784730	0.8687045	62485717	0.85	Willemain
0.863	0.8866964	0.8772976	64591255	0.86	Willemain
0.873	0.8949321	0.8859179	66836966	0.87	Willemain
0.883	0.9031510	0.8945442	69270895	0.88	Willemain
0.893	0.9111328	0.9029692	71898563	0.89	Willemain
0.914	0.9190149	0.9113021	74777592	0.90	Willemain
0.915	0.9268683	0.9196367	77940140	0.91	Willemain
0.923	0.9345870	0.9278612	81465324	0.92	Willemain
0.933	0.9422222	0.9360311	85427875	0.93	Willemain
0.943	0.9497273	0.9440953	89942949	0.94	Willemain
0.953	0.9570820	0.9520366	95185636	0.95	Willemain
0.963	0.9643109	0.9598772	101412257	0.96	Willemain
0.973	0.9715041	0.9677306	109031640	0.97	Willemain
0.983	0.9786343	0.9755900	118824750	0.98	Willemain
0.993	0.9859330	0.9837284	132943380	0.99	Willemain
0.754	0.7671551	0.7543221	43936948	0.75	QR
0.764	0.7768702	0.7642477	45233793	0.76	QR
0.774	0.7865767	0.7741792	46586465	0.77	QR
0.784	0.7964057	0.7842182	48011932	0.78	QR
0.794	0.8061412	0.7942257	49502054	0.79	QR
0.816	0.8158010	0.8041299	51059958	0.80	QR
0.817	0.8255758	0.8142065	52719836	0.81	QR
0.824	0.8351476	0.8240689	54455113	0.82	QR
0.834	0.8448005	0.8340405	56290511	0.83	QR

0.844	0.8543435	0.8439422	58253504	0.84	QR
0.854	0.8639401	0.8538882	60334198	0.85	QR
0.864	0.8734554	0.8637908	62571655	0.86	QR
0.874	0.8828659	0.8735918	64974262	0.87	QR
0.884	0.8922247	0.8833919	67566541	0.88	QR
0.894	0.9015667	0.8931912	70404823	0.89	QR
0.916	0.9108323	0.9029394	73514690	0.90	QR
0.917	0.9199892	0.9126201	76953836	0.91	QR
0.924	0.9290625	0.9222522	80815138	0.92	QR
0.934	0.9380243	0.9317964	85192921	0.93	QR
0.944	0.9469203	0.9413202	90245689	0.94	QR
0.954	0.9555617	0.9506256	96198018	0.95	QR
0.964	0.9641515	0.9599079	103381977	0.96	QR
0.974	0.9724961	0.9689804	112190302	0.97	QR
0.984	0.9799693	0.9771699	122622409	0.98	QR
0.994	0.9850041	0.9827712	132663548	0.99	QR
0.755	0.7692692	0.7571381	43938281	0.75	MLP
0.765	0.7780679	0.7660920	45097226	0.76	MLP
0.775	0.7869775	0.7751457	46296959	0.77	MLP
0.785	0.7958922	0.7842396	47555896	0.78	MLP
0.795	0.8048261	0.7933709	48866399	0.79	MLP
0.818	0.8137341	0.8024707	50237147	0.80	MLP
0.819	0.8227285	0.8116737	51680141	0.81	MLP
0.825	0.8315459	0.8207291	53191583	0.82	MLP
0.835	0.8406011	0.8300157	54784670	0.83	MLP
0.845	0.8494932	0.8391974	56482725	0.84	MLP
0.855	0.8584899	0.8484533	58268325	0.85	MLP
0.865	0.8674373	0.8577433	60195235	0.86	MLP
0.875	0.8764976	0.8670973	62236987	0.87	MLP
0.885	0.8854234	0.8764069	64452384	0.88	MLP
0.895	0.8944261	0.8857669	66851649	0.89	MLP
0.918	0.9033954	0.8951499	69467750	0.90	MLP
0.919	0.9124162	0.9046148	72352974	0.91	MLP

0.925	0.9214632	0.9141403	75580178	0.92	MLP
0.935	0.9304432	0.9236444	79214019	0.93	MLP
0.945	0.9395246	0.9332876	83410815	0.94	MLP
0.955	0.9485171	0.9428890	88340473	0.95	MLP
0.965	0.9575290	0.9525800	94365374	0.96	MLP
0.975	0.9666443	0.9624484	102031541	0.97	MLP
0.985	0.9756505	0.9722946	112192574	0.98	MLP
0.995	0.9832762	0.9807682	125137273	0.99	MLP
0.756	0.7810572	0.7596223	44858704	0.75	LSTM
0.766	0.7907242	0.7696558	46208624	0.76	LSTM
0.776	0.8002454	0.7795570	47597280	0.77	LSTM
0.786	0.8097337	0.7894819	49051164	0.78	LSTM
0.796	0.8191963	0.7993650	50580929	0.79	LSTM
0.820	0.8285261	0.8091617	52181247	0.80	LSTM
0.8110	0.8379734	0.8191437	53876163	0.81	LSTM
0.826	0.8473185	0.8289989	55660621	0.82	LSTM
0.836	0.8565389	0.8388375	57549172	0.83	LSTM
0.846	0.8657833	0.8486704	59555536	0.84	LSTM
0.856	0.8748942	0.8584143	61688287	0.85	LSTM
0.866	0.8839981	0.8682096	63980975	0.86	LSTM
0.876	0.8929203	0.8778574	66424947	0.87	LSTM
0.886	0.9018305	0.8874996	69085715	0.88	LSTM
0.896	0.9104753	0.8969189	71953216	0.89	LSTM
0.920	0.9192111	0.9064831	75102304	0.90	LSTM
0.9110	0.9277719	0.9159164	78573478	0.91	LSTM
0.926	0.9361696	0.9252244	82442717	0.92	LSTM
0.936	0.9444181	0.9344502	86788852	0.93	LSTM
0.946	0.9525138	0.9435604	91726249	0.94	LSTM
0.956	0.9603088	0.9523991	97404493	0.95	LSTM
0.966	0.9676687	0.9608464	104006744	0.96	LSTM
0.976	0.9745747	0.9688648	111697828	0.97	LSTM
0.986	0.9806791	0.9760366	120323556	0.98	LSTM
0.996	0.9849631	0.9811989	128997167	0.99	LSTM

0.757	0.7762886	0.7641197	44990067	0.75	LightGBM
0.767	0.7852212	0.7732110	46194497	0.76	LightGBM
0.777	0.7939690	0.7821401	47428598	0.77	LightGBM
0.787	0.8027587	0.7911333	48728191	0.78	LightGBM
0.797	0.8115432	0.8001213	50076824	0.79	LightGBM
0.827	0.8203414	0.8091520	51489504	0.80	LightGBM
0.8111	0.8292687	0.8183030	52986798	0.81	LightGBM
0.828	0.8379237	0.8272287	54545347	0.82	LightGBM
0.837	0.8467798	0.8363524	56195527	0.83	LightGBM
0.847	0.8554617	0.8453216	57943954	0.84	LightGBM
0.857	0.8643057	0.8544777	59798025	0.85	LightGBM
0.867	0.8730713	0.8635639	61787758	0.86	LightGBM
0.877	0.8817786	0.8726278	63903660	0.87	LightGBM
0.887	0.8904983	0.8817122	66198529	0.88	LightGBM
0.897	0.8991815	0.8907898	68679779	0.89	LightGBM
0.927	0.9079074	0.8999399	71400639	0.90	LightGBM
0.9111	0.9165818	0.9090653	74386932	0.91	LightGBM
0.928	0.9252623	0.9182504	77728583	0.92	LightGBM
0.937	0.9338691	0.9273732	81496091	0.93	LightGBM
0.947	0.9424723	0.9365344	85826467	0.94	LightGBM
0.957	0.9510094	0.9456700	90910155	0.95	LightGBM
0.967	0.9596220	0.9549413	97051693	0.96	LightGBM
0.977	0.9682396	0.9642714	104740078	0.97	LightGBM
0.987	0.9766659	0.9734812	114649946	0.98	LightGBM
0.997	0.9836525	0.9812118	126842666	0.99	LightGBM
0.758	0.7762886	0.7641197	44990067	0.75	RF
0.768	0.7852212	0.7732110	46194497	0.76	RF
0.778	0.7939690	0.7821401	47428598	0.77	RF
0.788	0.8027587	0.7911333	48728191	0.78	RF
0.798	0.8115432	0.8001213	50076824	0.79	RF
0.829	0.8203414	0.8091520	51489504	0.80	RF
0.8112	0.8292687	0.8183030	52986798	0.81	RF
0.8210	0.8379237	0.8272287	54545347	0.82	RF

0.838	0.8467798	0.8363524	56195527	0.83	RF
0.848	0.8554617	0.8453216	57943954	0.84	RF
0.858	0.8643057	0.8544777	59798025	0.85	RF
0.868	0.8730713	0.8635639	61787758	0.86	RF
0.878	0.8817786	0.8726278	63903660	0.87	RF
0.888	0.8904983	0.8817122	66198529	0.88	RF
0.898	0.8991815	0.8907898	68679779	0.89	RF
0.929	0.9079074	0.8999399	71400639	0.90	RF
0.9112	0.9165818	0.9090653	74386932	0.91	RF
0.9210	0.9252623	0.9182504	77728583	0.92	RF
0.938	0.9338691	0.9273732	81496091	0.93	RF
0.948	0.9424723	0.9365344	85826467	0.94	RF
0.958	0.9510094	0.9456700	90910155	0.95	RF
0.968	0.9596220	0.9549413	97051693	0.96	RF
0.978	0.9682396	0.9642714	104740078	0.97	RF
0.988	0.9766659	0.9734812	114649946	0.98	RF
0.998	0.9836525	0.9812118	126842666	0.99	RF

Table 13: IPM values for SIM2

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.7734392	0.7506230	30121950	0.75	Croston
0.76	0.7834493	0.7609711	31052328	0.76	Croston
0.77	0.7933339	0.7712286	32022249	0.77	Croston
0.78	0.8031410	0.7814656	33033949	0.78	Croston
0.79	0.8127568	0.7915096	34090417	0.79	Croston
0.8	0.8227712	0.8020162	35221779	0.80	Croston
0.81	0.8322854	0.8120373	36387481	0.81	Croston
0.82	0.8419696	0.8223075	37638906	0.82	Croston
0.83	0.8514241	0.8323029	38949549	0.83	Croston
0.84	0.8611721	0.8427099	40358422	0.84	Croston
0.85	0.8703950	0.8525840	41850122	0.85	Croston
0.86	0.8798418	0.8627251	43446526	0.86	Croston

0.87	0.8891434	0.8727577	45172927	0.87	Croston
0.88	0.8983514	0.8827953	47038137	0.88	Croston
0.89	0.9074671	0.8927422	49066961	0.89	Croston
0.9	0.9165969	0.9027875	51304620	0.90	Croston
0.91	0.9254899	0.9126540	53766695	0.91	Croston
0.92	0.9342878	0.9224744	56509143	0.92	Croston
0.93	0.9429518	0.9322373	59614606	0.93	Croston
0.94	0.9513941	0.9418225	63164591	0.94	Croston
0.95	0.9595037	0.9511267	67262659	0.95	Croston
0.96	0.9670272	0.9599223	71966832	0.96	Croston
0.97	0.9732783	0.9673901	77144466	0.97	Croston
0.98	0.9774876	0.9725629	82175566	0.98	Croston
0.99	0.9792701	0.9748286	85667213	0.99	Croston
0.751	0.7786980	0.7558367	30844404	0.75	SBA
0.761	0.7886454	0.7661720	31809899	0.76	SBA
0.771	0.7985167	0.7764257	32817133	0.77	SBA
0.781	0.8082953	0.7866959	33873332	0.78	SBA
0.791	0.8180043	0.7968587	34972473	0.79	SBA
0.810	0.8277624	0.8071852	36142031	0.80	SBA
0.811	0.8373763	0.8173008	37357624	0.81	SBA
0.821	0.8469456	0.8275007	38654094	0.82	SBA
0.831	0.8563242	0.8374694	40017721	0.83	SBA
0.841	0.8658440	0.8476450	41466323	0.84	SBA
0.851	0.8749839	0.8575089	43014235	0.85	SBA
0.861	0.8842620	0.8674903	44660816	0.86	SBA
0.871	0.8933760	0.8773695	46440005	0.87	SBA
0.881	0.9024576	0.8872678	48348782	0.88	SBA
0.891	0.9113564	0.8970448	50428971	0.89	SBA
0.910	0.9202034	0.9068256	52704191	0.90	SBA
0.911	0.9288281	0.9164007	55212462	0.91	SBA
0.921	0.9373184	0.9259118	57992966	0.92	SBA
0.931	0.9456530	0.9353348	61121982	0.93	SBA
0.941	0.9537351	0.9445252	64672367	0.94	SBA

0.951	0.9614604	0.9534332	68728671	0.95	SBA
0.961	0.9684522	0.9616282	73306884	0.96	SBA
0.971	0.9741418	0.9684635	78237588	0.97	SBA
0.981	0.9778891	0.9730805	82867496	0.98	SBA
0.991	0.9793255	0.9749078	85870569	0.99	SBA
0.752	0.7721011	0.7497451	29906492	0.75	DLP
0.762	0.7814869	0.7594121	30768160	0.76	DLP
0.772	0.7909622	0.7692159	31677233	0.77	DLP
0.782	0.8003352	0.7789788	32614858	0.78	DLP
0.792	0.8096011	0.7886127	33605259	0.79	DLP
0.812	0.8190319	0.7984931	34642020	0.80	DLP
0.813	0.8282796	0.8081870	35730107	0.81	DLP
0.822	0.8375585	0.8179614	36883295	0.82	DLP
0.832	0.8468012	0.8277358	38102262	0.83	DLP
0.842	0.8558670	0.8373748	39387131	0.84	DLP
0.852	0.8651571	0.8472450	40768650	0.85	DLP
0.862	0.8742393	0.8569773	42240649	0.86	DLP
0.872	0.8833292	0.8667197	43822009	0.87	DLP
0.882	0.8925125	0.8766398	45536790	0.88	DLP
0.892	0.9015101	0.8864104	47395087	0.89	DLP
0.912	0.9103865	0.8961388	49431691	0.90	DLP
0.913	0.9192532	0.9059158	51682619	0.91	DLP
0.922	0.9281215	0.9157451	54202358	0.92	DLP
0.932	0.9369352	0.9255796	57055432	0.93	DLP
0.942	0.9456900	0.9354511	60318872	0.94	DLP
0.952	0.9542884	0.9452268	64138599	0.95	DLP
0.962	0.9626022	0.9548172	68670081	0.96	DLP
0.972	0.9701646	0.9637303	73983599	0.97	DLP
0.982	0.9759253	0.9706857	79781247	0.98	DLP
0.992	0.9789727	0.9744644	84794589	0.99	DLP
0.753	0.7979339	0.7767975	32265859	0.75	Willemain
0.763	0.8064843	0.7857362	33153347	0.76	Willemain
0.773	0.8149731	0.7946429	34079332	0.77	Willemain

0.783	0.8233149	0.8034206	35045434	0.78	Willemain
0.793	0.8317315	0.8122737	36052001	0.79	Willemain
0.814	0.8398839	0.8209235	37107404	0.80	Willemain
0.815	0.8481409	0.8296475	38225619	0.81	Willemain
0.823	0.8561960	0.8382616	39391950	0.82	Willemain
0.833	0.8643491	0.8469524	40633114	0.83	Willemain
0.843	0.8723453	0.8555333	41940647	0.84	Willemain
0.853	0.8802615	0.8640758	43340342	0.85	Willemain
0.863	0.8881237	0.8725673	44827188	0.86	Willemain
0.873	0.8959379	0.8810485	46420457	0.87	Willemain
0.883	0.9036952	0.8895054	48138755	0.88	Willemain
0.893	0.9114588	0.8980160	50008126	0.89	Willemain
0.914	0.9189687	0.9063055	52016642	0.90	Willemain
0.915	0.9263919	0.9144915	54233516	0.91	Willemain
0.923	0.9337438	0.9226942	56673408	0.92	Willemain
0.933	0.9409545	0.9307537	59387820	0.93	Willemain
0.943	0.9480558	0.9387518	62428678	0.94	Willemain
0.953	0.9550607	0.9466797	65889497	0.95	Willemain
0.963	0.9620216	0.9546242	69889112	0.96	Willemain
0.973	0.9688411	0.9624767	74637368	0.97	Willemain
0.983	0.9757934	0.9705375	80581997	0.98	Willemain
0.993	0.9831190	0.9791439	89025356	0.99	Willemain
0.754	0.8548612	0.8375422	45093494	0.75	QR
0.764	0.8629740	0.8462496	46486328	0.76	QR
0.774	0.8707824	0.8546771	47931190	0.77	QR
0.784	0.8783978	0.8628810	49419711	0.78	QR
0.794	0.8858104	0.8709610	50984129	0.79	QR
0.816	0.8931884	0.8789834	52600509	0.80	QR
0.817	0.9002238	0.8867030	54275722	0.81	QR
0.824	0.9070945	0.8942348	56017038	0.82	QR
0.834	0.9137435	0.9015659	57815257	0.83	QR
0.844	0.9202007	0.9086926	59677752	0.84	QR
0.854	0.9263261	0.9155138	61588677	0.85	QR

0.864	0.9322037	0.9220616	63563293	0.86	QR
0.874	0.9377317	0.9282925	65591887	0.87	QR
0.884	0.9429651	0.9341911	67674637	0.88	QR
0.894	0.9478852	0.9397907	69803317	0.89	QR
0.916	0.9525468	0.9450926	71986997	0.90	QR
0.917	0.9568289	0.9499971	74182125	0.91	QR
0.924	0.9607623	0.9545233	76371099	0.92	QR
0.934	0.9643205	0.9586252	78493640	0.93	QR
0.944	0.9672862	0.9621023	80502607	0.94	QR
0.954	0.9696622	0.9649213	82313444	0.95	QR
0.964	0.9714094	0.9670349	83805159	0.96	QR
0.974	0.9725118	0.9683613	84893058	0.97	QR
0.984	0.9730613	0.9690104	85513928	0.98	QR
0.994	0.9731770	0.9691536	85712795	0.99	QR
0.755	0.7679743	0.7458425	29429718	0.75	MLP
0.765	0.7781350	0.7563274	30325278	0.76	MLP
0.775	0.7881142	0.7666691	31267554	0.77	MLP
0.785	0.7981095	0.7770071	32246083	0.78	MLP
0.795	0.8078955	0.7872045	33269491	0.79	MLP
0.818	0.8179177	0.7976727	34363196	0.80	MLP
0.819	0.8277943	0.8080081	35492762	0.81	MLP
0.825	0.8373122	0.8180586	36696758	0.82	MLP
0.835	0.8473200	0.8285639	37969772	0.83	MLP
0.845	0.8568882	0.8386910	39326433	0.84	MLP
0.855	0.8665407	0.8489523	40775124	0.85	MLP
0.865	0.8761096	0.8591727	42331736	0.86	MLP
0.875	0.8855213	0.8692934	43996553	0.87	MLP
0.885	0.8948348	0.8793502	45806164	0.88	MLP
0.895	0.9039983	0.8892945	47780048	0.89	MLP
0.918	0.9132451	0.8994012	49948210	0.90	MLP
0.919	0.9221422	0.9092050	52346978	0.91	MLP
0.925	0.9309851	0.9190190	55024601	0.92	MLP
0.935	0.9396366	0.9286899	58043270	0.93	MLP

0.945	0.9480125	0.9381538	61461310	0.94	MLP
0.955	0.9560298	0.9473212	65323000	0.95	MLP
0.965	0.9635367	0.9560286	69642628	0.96	MLP
0.975	0.9700924	0.9637559	74356107	0.97	MLP
0.985	0.9753704	0.9700724	79258678	0.98	MLP
0.995	0.9786032	0.9740222	83830841	0.99	MLP
0.756	0.7800323	0.7689542	33258734	0.75	LSTM
0.766	0.7925745	0.7817538	34314804	0.76	LSTM
0.776	0.8050678	0.7944884	35420910	0.77	LSTM
0.786	0.8170601	0.8067432	36563134	0.78	LSTM
0.796	0.8291345	0.8191205	37767676	0.79	LSTM
0.820	0.8411632	0.8314963	39032626	0.80	LSTM
0.8110	0.8527397	0.8433938	40367390	0.81	LSTM
0.826	0.8641216	0.8551364	41756588	0.82	LSTM
0.836	0.8754236	0.8668375	43247513	0.83	LSTM
0.846	0.8862378	0.8780619	44798658	0.84	LSTM
0.856	0.8967369	0.8889806	46462964	0.85	LSTM
0.866	0.9071293	0.8998580	48234576	0.86	LSTM
0.876	0.9169026	0.9100991	50115337	0.87	LSTM
0.886	0.9264457	0.9201571	52119492	0.88	LSTM
0.896	0.9354749	0.9296926	54289378	0.89	LSTM
0.920	0.9441483	0.9388929	56602570	0.90	LSTM
0.9110	0.9524408	0.9477256	59103538	0.91	LSTM
0.926	0.9600420	0.9558571	61785933	0.92	LSTM
0.936	0.9670912	0.9634571	64675928	0.93	LSTM
0.946	0.9735336	0.9704474	67782153	0.94	LSTM
0.956	0.9791851	0.9766242	71065986	0.95	LSTM
0.966	0.9839443	0.9818740	74516871	0.96	LSTM
0.976	0.9878332	0.9862025	77943260	0.97	LSTM
0.986	0.9905127	0.9892215	81050177	0.98	LSTM
0.996	0.9919008	0.9908056	83304288	0.99	LSTM
0.757	0.7934391	0.7735172	33433219	0.75	LightGBM
0.767	0.8027891	0.7832290	34442956	0.76	LightGBM

0.777	0.8122043	0.7929472	35498297	0.77	LightGBM
0.787	0.8215161	0.8026820	36601382	0.78	LightGBM
0.797	0.8306075	0.8121548	37749801	0.79	LightGBM
0.827	0.8399048	0.8218704	38959709	0.80	LightGBM
0.8111	0.8486324	0.8310161	40219457	0.81	LightGBM
0.828	0.8575872	0.8404864	41546917	0.82	LightGBM
0.837	0.8663898	0.8497509	42954461	0.83	LightGBM
0.847	0.8750841	0.8589976	44425497	0.84	LightGBM
0.857	0.8836362	0.8681292	45991273	0.85	LightGBM
0.867	0.8919949	0.8770743	47647170	0.86	LightGBM
0.877	0.9002805	0.8860206	49423611	0.87	LightGBM
0.887	0.9084203	0.8948328	51313598	0.88	LightGBM
0.897	0.9163689	0.9034789	53343521	0.89	LightGBM
0.927	0.9242755	0.9121390	55537920	0.90	LightGBM
0.9111	0.9318860	0.9205499	57911663	0.91	LightGBM
0.928	0.9393460	0.9288356	60491391	0.92	LightGBM
0.937	0.9465356	0.9368861	63296431	0.93	LightGBM
0.947	0.9535051	0.9447220	66365835	0.94	LightGBM
0.957	0.9600844	0.9522052	69715939	0.95	LightGBM
0.967	0.9662383	0.9593076	73338837	0.96	LightGBM
0.977	0.9716232	0.9655896	77194609	0.97	LightGBM
0.987	0.9759321	0.9707292	81073317	0.98	LightGBM
0.997	0.9784623	0.9738369	84532894	0.99	LightGBM
0.758	0.7868137	0.7642859	31596342	0.75	RF
0.768	0.7963612	0.7741919	32562416	0.76	RF
0.778	0.8060163	0.7842373	33581384	0.77	RF
0.788	0.8154367	0.7941075	34639510	0.78	RF
0.798	0.8248611	0.8039458	35754610	0.79	RF
0.829	0.8342336	0.8138416	36932108	0.80	RF
0.8112	0.8434690	0.8236122	38156565	0.81	RF
0.8210	0.8525917	0.8332971	39458993	0.82	RF
0.838	0.8616822	0.8429770	40830026	0.83	RF
0.848	0.8705407	0.8524549	42291750	0.84	RF

0.858	0.8794923	0.8620824	43840349	0.85	RF
0.868	0.8881598	0.8714453	45494962	0.86	RF
0.878	0.8967872	0.8808006	47267254	0.87	RF
0.888	0.9053413	0.8901737	49173983	0.88	RF
0.898	0.9136860	0.8993335	51225961	0.89	RF
0.929	0.9219028	0.9084587	53451614	0.90	RF
0.9112	0.9298652	0.9173322	55863290	0.91	RF
0.9210	0.9376190	0.9260332	58500987	0.92	RF
0.938	0.9451538	0.9345975	61392625	0.93	RF
0.948	0.9523533	0.9428436	64585206	0.94	RF
0.958	0.9592372	0.9507932	68103720	0.95	RF
0.968	0.9655485	0.9582355	71954560	0.96	RF
0.978	0.9711746	0.9649686	76083061	0.97	RF
0.988	0.9758178	0.9705733	80319265	0.98	RF
0.998	0.9787146	0.9741475	84230049	0.99	RF

Table 14: IPM values for SIM3

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.7764167	0.7712013	31909257	0.75	Croston
0.76	0.7859988	0.7808508	32623913	0.76	Croston
0.77	0.7954365	0.7903327	33367340	0.77	Croston
0.78	0.8050072	0.7999813	34141501	0.78	Croston
0.79	0.8143336	0.8093938	34941166	0.79	Croston
0.8	0.8237397	0.8188920	35768473	0.80	Croston
0.81	0.8331264	0.8283433	36638853	0.81	Croston
0.82	0.8424798	0.8378063	37551626	0.82	Croston
0.83	0.8517458	0.8471991	38513701	0.83	Croston
0.84	0.8610118	0.8565685	39520696	0.84	Croston
0.85	0.8702430	0.8659405	40578409	0.85	Croston
0.86	0.8794284	0.8752819	41718506	0.86	Croston
0.87	0.8886272	0.8846234	42922362	0.87	Croston
0.88	0.8976902	0.8938657	44211578	0.88	Croston

0.89	0.9067094	0.9030621	45605771	0.89	Croston
0.9	0.9158223	0.9123585	47121343	0.90	Croston
0.91	0.9246198	0.9213757	48780882	0.91	Croston
0.92	0.9335335	0.9305126	50619271	0.92	Croston
0.93	0.9423695	0.9395901	52694090	0.93	Croston
0.94	0.9510337	0.9485307	55049540	0.94	Croston
0.95	0.9595865	0.9573705	57809151	0.95	Croston
0.96	0.9680613	0.9661723	61152449	0.96	Croston
0.97	0.9763094	0.9747788	65395646	0.97	Croston
0.98	0.9842215	0.9830961	71157310	0.98	Croston
0.99	0.9905401	0.9897993	79021834	0.99	Croston
0.751	0.7715539	0.7660419	31522100	0.75	SBA
0.761	0.7812129	0.7757427	32235660	0.76	SBA
0.771	0.7909262	0.7855084	32980310	0.77	SBA
0.781	0.8006677	0.7953407	33759996	0.78	SBA
0.791	0.8102762	0.8050524	34557734	0.79	SBA
0.810	0.8196747	0.8145487	35379848	0.80	SBA
0.811	0.8292887	0.8242333	36257131	0.81	SBA
0.821	0.8390436	0.8341107	37179642	0.82	SBA
0.831	0.8484619	0.8436647	38131868	0.83	SBA
0.841	0.8578139	0.8531214	39139565	0.84	SBA
0.851	0.8673125	0.8627862	40214595	0.85	SBA
0.861	0.8766907	0.8723221	41346428	0.86	SBA
0.871	0.8860910	0.8818743	42555603	0.87	SBA
0.881	0.8954217	0.8913995	43857270	0.88	SBA
0.891	0.9046510	0.9008157	45254618	0.89	SBA
0.910	0.9139054	0.9102715	46778274	0.90	SBA
0.911	0.9230122	0.9196201	48451923	0.91	SBA
0.921	0.9321340	0.9289760	50297960	0.92	SBA
0.931	0.9411951	0.9383021	52388689	0.93	SBA
0.941	0.9500611	0.9474562	54765529	0.94	SBA
0.951	0.9588657	0.9565688	57548350	0.95	SBA
0.961	0.9675399	0.9655932	60920821	0.96	SBA

0.971	0.9759374	0.9743608	65207888	0.97	SBA
0.981	0.9840616	0.9829114	71043416	0.98	SBA
0.991	0.9905248	0.9897795	79014784	0.99	SBA
0.752	0.8130700	0.8084985	35334420	0.75	DLP
0.762	0.8214010	0.8168960	36068986	0.76	DLP
0.772	0.8296141	0.8251710	36834676	0.77	DLP
0.782	0.8377148	0.8333369	37614282	0.78	DLP
0.792	0.8458380	0.8415516	38425623	0.79	DLP
0.812	0.8537729	0.8495644	39263120	0.80	DLP
0.813	0.8617147	0.8576088	40148451	0.81	DLP
0.822	0.8696389	0.8656469	41062952	0.82	DLP
0.832	0.8774560	0.8735742	42019827	0.83	DLP
0.842	0.8854000	0.8816248	43025877	0.84	DLP
0.852	0.8928940	0.8892540	44088536	0.85	DLP
0.862	0.9006480	0.8971389	45211824	0.86	DLP
0.872	0.9083729	0.9049951	46405022	0.87	DLP
0.882	0.9158215	0.9126017	47675371	0.88	DLP
0.892	0.9233365	0.9202696	49044104	0.89	DLP
0.912	0.9307104	0.9278285	50520906	0.90	DLP
0.913	0.9379783	0.9352748	52133898	0.91	DLP
0.922	0.9452327	0.9427265	53909330	0.92	DLP
0.932	0.9523637	0.9500530	55887788	0.93	DLP
0.942	0.9594068	0.9573191	58143400	0.94	DLP
0.952	0.9663363	0.9644889	60750075	0.95	DLP
0.962	0.9731575	0.9715722	63867318	0.96	DLP
0.972	0.9797934	0.9784979	67763659	0.97	DLP
0.982	0.9859955	0.9849948	72890050	0.98	DLP
0.992	0.9909160	0.9902091	79654657	0.99	DLP
0.753	0.8244658	0.8187290	35625098	0.75	Willemain
0.763	0.8327957	0.8271652	36396015	0.76	Willemain
0.773	0.8410564	0.8355770	37193164	0.77	Willemain
0.783	0.8494160	0.8440835	38037258	0.78	Willemain
0.793	0.8575975	0.8524044	38894385	0.79	Willemain

0.814	0.8656280	0.8605740	39791866	0.80	Willemain
0.815	0.8735033	0.8686039	40723842	0.81	Willemain
0.823	0.8814080	0.8766844	41703306	0.82	Willemain
0.833	0.8892051	0.8846549	42721795	0.83	Willemain
0.843	0.8968427	0.8924705	43804703	0.84	Willemain
0.853	0.9044465	0.9002590	44941344	0.85	Willemain
0.863	0.9118143	0.9078423	46133988	0.86	Willemain
0.873	0.9191000	0.9153291	47426036	0.87	Willemain
0.883	0.9263270	0.9227700	48788067	0.88	Willemain
0.893	0.9334073	0.9300938	50256660	0.89	Willemain
0.914	0.9403207	0.9372158	51849866	0.90	Willemain
0.915	0.9471867	0.9443469	53588935	0.91	Willemain
0.923	0.9536567	0.9510483	55508008	0.92	Willemain
0.933	0.9602161	0.9578767	57659856	0.93	Willemain
0.943	0.9663757	0.9643024	60099289	0.94	Willemain
0.953	0.9724050	0.9706102	62939762	0.95	Willemain
0.963	0.9782955	0.9768090	66346125	0.96	Willemain
0.973	0.9838529	0.9826745	70619649	0.97	Willemain
0.983	0.9891549	0.9883050	76362326	0.98	Willemain
0.993	0.9941252	0.9936166	85313109	0.99	Willemain
0.754	0.7595192	0.7531615	30583248	0.75	QR
0.764	0.7696027	0.7633226	31294342	0.76	QR
0.774	0.7797156	0.7735098	32037702	0.77	QR
0.784	0.7896421	0.7835529	32799958	0.78	QR
0.794	0.7998942	0.7939176	33616720	0.79	QR
0.816	0.8100799	0.8042408	34457835	0.80	QR
0.817	0.8202560	0.8145064	35335690	0.81	QR
0.824	0.8302237	0.8246179	36256838	0.82	QR
0.834	0.8402141	0.8347682	37214308	0.83	QR
0.844	0.8504787	0.8452094	38251194	0.84	QR
0.854	0.8602916	0.8551822	39326051	0.85	QR
0.864	0.8701919	0.8652740	40473024	0.86	QR
0.874	0.8802671	0.8755648	41708123	0.87	QR

0.884	0.8900763	0.8855655	43021923	0.88	QR
0.894	0.9000195	0.8957536	44454877	0.89	QR
0.916	0.9097526	0.9057418	45996826	0.90	QR
0.917	0.9194142	0.9156723	47706495	0.91	QR
0.924	0.9290227	0.9255739	49591097	0.92	QR
0.934	0.9386534	0.9355144	51730693	0.93	QR
0.944	0.9480433	0.9452251	54157830	0.94	QR
0.954	0.9573362	0.9548682	57023734	0.95	QR
0.964	0.9663990	0.9643214	60497021	0.96	QR
0.974	0.9752900	0.9736448	64913859	0.97	QR
0.984	0.9837543	0.9825629	70957428	0.98	QR
0.994	0.9904968	0.9897443	79176225	0.99	QR
0.755	0.7772962	0.7721137	32077544	0.75	MLP
0.765	0.7867277	0.7816128	32787644	0.76	MLP
0.775	0.7960367	0.7909830	33530219	0.77	MLP
0.785	0.8054302	0.8004496	34294669	0.78	MLP
0.795	0.8147148	0.8098163	35089273	0.79	MLP
0.818	0.8241672	0.8193487	35925035	0.80	MLP
0.819	0.8333822	0.8286703	36791220	0.81	MLP
0.825	0.8427077	0.8381036	37696652	0.82	MLP
0.835	0.8518436	0.8473540	38648313	0.83	MLP
0.845	0.8611293	0.8567558	39658142	0.84	MLP
0.855	0.8702393	0.8660009	40713010	0.85	MLP
0.865	0.8793873	0.8753117	41836791	0.86	MLP
0.875	0.8886169	0.8846828	43036684	0.87	MLP
0.885	0.8976246	0.8938540	44316703	0.88	MLP
0.895	0.9066436	0.9030612	45710048	0.89	MLP
0.918	0.9155600	0.9121621	47204076	0.90	MLP
0.919	0.9245195	0.9213369	48864332	0.91	MLP
0.925	0.9333942	0.9304361	50681853	0.92	MLP
0.935	0.9422632	0.9395424	52738839	0.93	MLP
0.945	0.9509346	0.9484758	55072592	0.94	MLP
0.955	0.9595790	0.9574083	57817691	0.95	MLP

0.965	0.9680745	0.9662363	61133569	0.96	MLP
0.975	0.9764136	0.9749328	65343292	0.97	MLP
0.985	0.9843404	0.9832528	71061714	0.98	MLP
0.995	0.9906547	0.9899308	78910381	0.99	MLP
0.756	0.7800323	0.7689542	33258734	0.75	LSTM
0.766	0.7925745	0.7817538	34314804	0.76	LSTM
0.776	0.8050678	0.7944884	35420910	0.77	LSTM
0.786	0.8170601	0.8067432	36563134	0.78	LSTM
0.796	0.8291345	0.8191205	37767676	0.79	LSTM
0.820	0.8411632	0.8314963	39032626	0.80	LSTM
0.8110	0.8527397	0.8433938	40367390	0.81	LSTM
0.826	0.8641216	0.8551364	41756588	0.82	LSTM
0.836	0.8754236	0.8668375	43247513	0.83	LSTM
0.846	0.8862378	0.8780619	44798658	0.84	LSTM
0.856	0.8967369	0.8889806	46462964	0.85	LSTM
0.866	0.9071293	0.8998580	48234576	0.86	LSTM
0.876	0.9169026	0.9100991	50115337	0.87	LSTM
0.886	0.9264457	0.9201571	52119492	0.88	LSTM
0.896	0.9354749	0.9296926	54289378	0.89	LSTM
0.920	0.9441483	0.9388929	56602570	0.90	LSTM
0.9110	0.9524408	0.9477256	59103538	0.91	LSTM
0.926	0.9600420	0.9558571	61785933	0.92	LSTM
0.936	0.9670912	0.9634571	64675928	0.93	LSTM
0.946	0.9735336	0.9704474	67782153	0.94	LSTM
0.956	0.9791851	0.9766242	71065986	0.95	LSTM
0.966	0.9839443	0.9818740	74516871	0.96	LSTM
0.976	0.9878332	0.9862025	77943260	0.97	LSTM
0.986	0.9905127	0.9892215	81050177	0.98	LSTM
0.996	0.9919008	0.9908056	83304288	0.99	LSTM
0.757	0.7781139	0.7721524	32105916	0.75	LightGBM
0.767	0.7874768	0.7816011	32822084	0.76	LightGBM
0.777	0.7969502	0.7911848	33570929	0.77	LightGBM
0.787	0.8063803	0.8007199	34343774	0.78	LightGBM

0.797	0.8157910	0.8102306	35150209	0.79	LightGBM
0.827	0.8251493	0.8197063	35985024	0.80	LightGBM
0.8111	0.8344994	0.8291945	36860952	0.81	LightGBM
0.828	0.8437718	0.8385882	37772294	0.82	LightGBM
0.837	0.8531093	0.8480782	38736398	0.83	LightGBM
0.847	0.8623604	0.8574827	39749551	0.84	LightGBM
0.857	0.8715732	0.8668728	40821839	0.85	LightGBM
0.867	0.8806427	0.8761196	41954052	0.86	LightGBM
0.877	0.8898594	0.8855250	43167371	0.87	LightGBM
0.887	0.8989370	0.8948070	44468150	0.88	LightGBM
0.897	0.9079335	0.9040349	45861533	0.89	LightGBM
0.927	0.9169508	0.9132808	47387181	0.90	LightGBM
0.9111	0.9258630	0.9224403	49056316	0.91	LightGBM
0.928	0.9347422	0.9315899	50907774	0.92	LightGBM
0.937	0.9433968	0.9405152	52976540	0.93	LightGBM
0.947	0.9521210	0.9495486	55359804	0.94	LightGBM
0.957	0.9605889	0.9583396	58133671	0.95	LightGBM
0.967	0.9688799	0.9669812	61492958	0.96	LightGBM
0.977	0.9770392	0.9755327	65752224	0.97	LightGBM
0.987	0.9847682	0.9836662	71518743	0.98	LightGBM
0.997	0.9907316	0.9900073	79246300	0.99	LightGBM
0.758	0.7806374	0.7751582	32354705	0.75	RF
0.768	0.7898578	0.7844401	33067419	0.76	RF
0.778	0.7992334	0.7939104	33812532	0.77	RF
0.788	0.8085333	0.8032788	34588881	0.78	RF
0.798	0.8178193	0.8126698	35389716	0.79	RF
0.829	0.8270418	0.8219896	36220271	0.80	RF
0.8112	0.8362808	0.8313409	37090688	0.81	RF
0.8210	0.8454281	0.8406085	38006435	0.82	RF
0.838	0.8546065	0.8499013	38962476	0.83	RF
0.848	0.8636784	0.8591274	39967465	0.84	RF
0.858	0.8726749	0.8682653	41030407	0.85	RF
0.868	0.8818019	0.8775581	42161397	0.86	RF

0.878	0.8907300	0.8866581	43361488	0.87	RF
0.888	0.8996337	0.8957401	44652267	0.88	RF
0.898	0.9086234	0.9049320	46046752	0.89	RF
0.929	0.9173228	0.9138357	47546099	0.90	RF
0.9112	0.9262527	0.9229817	49206705	0.91	RF
0.9210	0.9349564	0.9319322	51039532	0.92	RF
0.938	0.9436701	0.9409052	53095331	0.93	RF
0.948	0.9522183	0.9497260	55444357	0.94	RF
0.958	0.9606721	0.9584874	58190478	0.95	RF
0.968	0.9689388	0.9670938	61513880	0.96	RF
0.978	0.9770174	0.9755345	65725504	0.97	RF
0.988	0.9847491	0.9836626	71436971	0.98	RF
0.998	0.9907489	0.9900263	79161129	0.99	RF

Table 15: IPM values for SIM4

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.7742640	0.7670440	22355860	0.75	Croston
0.76	0.7861326	0.7789816	22957806	0.76	Croston
0.77	0.7980579	0.7909848	23578620	0.77	Croston
0.78	0.8098851	0.8029249	24233149	0.78	Croston
0.79	0.8215289	0.8147067	24906352	0.79	Croston
0.8	0.8330974	0.8263908	25618805	0.80	Croston
0.81	0.8447542	0.8382456	26370994	0.81	Croston
0.82	0.8561924	0.8498531	27157550	0.82	Croston
0.83	0.8671939	0.8610229	27981358	0.83	Croston
0.84	0.8784413	0.8725562	28863124	0.84	Croston
0.85	0.8893741	0.8837075	29804210	0.85	Croston
0.86	0.8999892	0.8945929	30799496	0.86	Croston
0.87	0.9104557	0.9053658	31867979	0.87	Croston
0.88	0.9206338	0.9158370	33031255	0.88	Croston
0.89	0.9303775	0.9259274	34275678	0.89	Croston
0.9	0.9398800	0.9357854	35662787	0.90	Croston

0.91	0.9489046	0.9451871	37163950	0.91	Croston
0.92	0.9575069	0.9542167	38863830	0.92	Croston
0.93	0.9656202	0.9627603	40765670	0.93	Croston
0.94	0.9730645	0.9706685	42955812	0.94	Croston
0.95	0.9797143	0.9777976	45466751	0.95	Croston
0.96	0.9851446	0.9836830	48295049	0.96	Croston
0.97	0.9888856	0.9877904	51222605	0.97	Croston
0.98	0.9907451	0.9898676	53705061	0.98	Croston
0.99	0.9911758	0.9903658	55100241	0.99	Croston
0.751	0.7729708	0.7653798	22296521	0.75	SBA
0.761	0.7850935	0.7776104	22904245	0.76	SBA
0.771	0.7971521	0.7897459	23536064	0.77	SBA
0.781	0.8091776	0.8019085	24200580	0.78	SBA
0.791	0.8209859	0.8138672	24886896	0.79	SBA
0.810	0.8326658	0.8256713	25607517	0.80	SBA
0.811	0.8444856	0.8377103	26372447	0.81	SBA
0.821	0.8561978	0.8496318	27177902	0.82	SBA
0.831	0.8672130	0.8608090	28008030	0.83	SBA
0.841	0.8786695	0.8725871	28916067	0.84	SBA
0.851	0.8897523	0.8839016	29867216	0.85	SBA
0.861	0.9004287	0.8948414	30885521	0.86	SBA
0.871	0.9110759	0.9058295	31973967	0.87	SBA
0.881	0.9212533	0.9163316	33161406	0.88	SBA
0.891	0.9311280	0.9265555	34431606	0.89	SBA
0.910	0.9406017	0.9364222	35838299	0.90	SBA
0.911	0.9497083	0.9459339	37373138	0.91	SBA
0.921	0.9582888	0.9549561	39101284	0.92	SBA
0.931	0.9662793	0.9633983	41028898	0.93	SBA
0.941	0.9736955	0.9713027	43245912	0.94	SBA
0.951	0.9802405	0.9783355	45773609	0.95	SBA
0.961	0.9854873	0.9840391	48590585	0.96	SBA
0.971	0.9890603	0.9879758	51458087	0.97	SBA
0.981	0.9907795	0.9899059	53833517	0.98	SBA

0.991	0.9911768	0.9903671	55132431	0.99	SBA
0.752	0.7953476	0.7887976	23580543	0.75	DLP
0.762	0.8060370	0.7995420	24164067	0.76	DLP
0.772	0.8167128	0.8102605	24775697	0.77	DLP
0.782	0.8273923	0.8211151	25402273	0.78	DLP
0.792	0.8377606	0.8316011	26061037	0.79	DLP
0.812	0.8481112	0.8420526	26744390	0.80	DLP
0.813	0.8585720	0.8526598	27473111	0.81	DLP
0.822	0.8686505	0.8629183	28229000	0.82	DLP
0.832	0.8786072	0.8730446	29018707	0.83	DLP
0.842	0.8886429	0.8833291	29867475	0.84	DLP
0.852	0.8983077	0.8931747	30760996	0.85	DLP
0.862	0.9076501	0.9027607	31707658	0.86	DLP
0.872	0.9170276	0.9124097	32728613	0.87	DLP
0.882	0.9260775	0.9217051	33820982	0.88	DLP
0.892	0.9347834	0.9307359	35001185	0.89	DLP
0.912	0.9434276	0.9396727	36297433	0.90	DLP
0.913	0.9514560	0.9480717	37705908	0.91	DLP
0.922	0.9593093	0.9562617	39291010	0.92	DLP
0.932	0.9666808	0.9640178	41061010	0.93	DLP
0.942	0.9735185	0.9712632	43083494	0.94	DLP
0.952	0.9797036	0.9778743	45411804	0.95	DLP
0.962	0.9849115	0.9834790	48059288	0.96	DLP
0.972	0.9886566	0.9875542	50870857	0.97	DLP
0.982	0.9906322	0.9897464	53419521	0.98	DLP
0.992	0.9911642	0.9903522	55004076	0.99	DLP
0.753	0.8326600	0.8255971	25501850	0.75	Willemain
0.763	0.8411109	0.8341988	26038226	0.76	Willemain
0.773	0.8493537	0.8425966	26587798	0.77	Willemain
0.783	0.8575546	0.8509980	27158380	0.78	Willemain
0.793	0.8655356	0.8591695	27749299	0.79	Willemain
0.814	0.8736113	0.8674424	28368427	0.80	Willemain
0.815	0.8814504	0.8754779	29011041	0.81	Willemain

0.823	0.8891759	0.8834416	29674466	0.82	Willemain
0.833	0.8967927	0.8912793	30368382	0.83	Willemain
0.843	0.9043616	0.8990527	31113195	0.84	Willemain
0.853	0.9115454	0.9064996	31880630	0.85	Willemain
0.863	0.9188802	0.9140924	32694307	0.86	Willemain
0.873	0.9258353	0.9212847	33563779	0.87	Willemain
0.883	0.9327715	0.9284967	34487736	0.88	Willemain
0.893	0.9395985	0.9356271	35481247	0.89	Willemain
0.914	0.9461517	0.9424583	36560420	0.90	Willemain
0.915	0.9524982	0.9491115	37728002	0.91	Willemain
0.923	0.9586915	0.9556237	39019093	0.92	Willemain
0.933	0.9646450	0.9618899	40454068	0.93	Willemain
0.943	0.9704494	0.9680423	42087829	0.94	Willemain
0.953	0.9759269	0.9738733	43976075	0.95	Willemain
0.963	0.9812545	0.9795731	46237330	0.96	Willemain
0.973	0.9862293	0.9849244	49039189	0.97	Willemain
0.983	0.9908760	0.9899578	52786186	0.98	Willemain
0.993	0.9951729	0.9946352	58594299	0.99	Willemain
0.754	0.7744098	0.7666991	22427970	0.75	QR
0.764	0.7865189	0.7789013	23048931	0.76	QR
0.774	0.7985573	0.7910540	23687673	0.77	QR
0.784	0.8103755	0.8030856	24359547	0.78	QR
0.794	0.8221426	0.8150504	25060157	0.79	QR
0.816	0.8340170	0.8270832	25800385	0.80	QR
0.817	0.8456978	0.8389912	26575831	0.81	QR
0.824	0.8571487	0.8506914	27394605	0.82	QR
0.834	0.8683029	0.8620998	28247059	0.83	QR
0.844	0.8795137	0.8736084	29168821	0.84	QR
0.854	0.8905158	0.8849019	30141135	0.85	QR
0.864	0.9011035	0.8957873	31176276	0.86	QR
0.874	0.9113867	0.9064514	32284950	0.87	QR
0.884	0.9214584	0.9168459	33495138	0.88	QR
0.894	0.9311240	0.9269413	34786222	0.89	QR

0.916	0.9405366	0.9367177	36220954	0.90	QR
0.917	0.9492881	0.9459240	37774261	0.91	QR
0.924	0.9577200	0.9547755	39525349	0.92	QR
0.934	0.9655068	0.9630051	41474996	0.93	QR
0.944	0.9726406	0.9706141	43679863	0.94	QR
0.954	0.9788918	0.9773266	46169530	0.95	QR
0.964	0.9839678	0.9828336	48906198	0.96	QR
0.974	0.9874422	0.9866356	51651097	0.97	QR
0.984	0.9891572	0.9885421	53868113	0.98	QR
0.994	0.9895338	0.9889699	55040130	0.99	QR
0.755	0.7719820	0.7645044	22254120	0.75	MLP
0.765	0.7838434	0.7764161	22844176	0.76	MLP
0.775	0.7960008	0.7886900	23467025	0.77	MLP
0.785	0.8075847	0.8004557	24105097	0.78	MLP
0.795	0.8190645	0.8120558	24766071	0.79	MLP
0.818	0.8306155	0.8237808	25464731	0.80	MLP
0.819	0.8423115	0.8357159	26207597	0.81	MLP
0.825	0.8538254	0.8474100	26988910	0.82	MLP
0.835	0.8647681	0.8585538	27788754	0.83	MLP
0.845	0.8759857	0.8700611	28654713	0.84	MLP
0.855	0.8871939	0.8815289	29588967	0.85	MLP
0.865	0.8976189	0.8922116	30556194	0.86	MLP
0.875	0.9082865	0.9032144	31616930	0.87	MLP
0.885	0.9185306	0.9137685	32751981	0.88	MLP
0.895	0.9283953	0.9239813	33979800	0.89	MLP
0.918	0.9380797	0.9340346	35330714	0.90	MLP
0.919	0.9471978	0.9435526	36808587	0.91	MLP
0.925	0.9560879	0.9528344	38481143	0.92	MLP
0.935	0.9642261	0.9613941	40346367	0.93	MLP
0.945	0.9718143	0.9694456	42488752	0.94	MLP
0.955	0.9785740	0.9766737	44948921	0.95	MLP
0.965	0.9842349	0.9827755	47707378	0.96	MLP
0.975	0.9882718	0.9871635	50559875	0.97	MLP

0.985	0.9904740	0.9895807	53124631	0.98	MLP
0.995	0.9911402	0.9903263	54850228	0.99	MLP
0.756	0.8524185	0.8406716	27934481	0.75	LSTM
0.766	0.8638138	0.8524171	28832926	0.76	LSTM
0.776	0.8749118	0.8639847	29756988	0.77	LSTM
0.786	0.8852849	0.8748795	30720936	0.78	LSTM
0.796	0.8952637	0.8853520	31724192	0.79	LSTM
0.820	0.9051474	0.8957680	32767952	0.80	LSTM
0.8110	0.9145091	0.9056687	33855279	0.81	LSTM
0.826	0.9233015	0.9150359	34982705	0.82	LSTM
0.836	0.9316930	0.9239809	36161844	0.83	LSTM
0.846	0.9394695	0.9323177	37392693	0.84	LSTM
0.856	0.9469582	0.9404120	38670043	0.85	LSTM
0.866	0.9538404	0.9478496	40002334	0.86	LSTM
0.876	0.9600787	0.9546771	41367213	0.87	LSTM
0.886	0.9658505	0.9610277	42782352	0.88	LSTM
0.896	0.9710195	0.9667216	44235722	0.89	LSTM
0.920	0.9755433	0.9717588	45716577	0.90	LSTM
0.9110	0.9794334	0.9761373	47194499	0.91	LSTM
0.926	0.9827058	0.9798410	48651922	0.92	LSTM
0.936	0.9854698	0.9830112	50054415	0.93	LSTM
0.946	0.9875611	0.9854399	51345089	0.94	LSTM
0.956	0.9890124	0.9871372	52478699	0.95	LSTM
0.966	0.9899500	0.9882606	53420685	0.96	LSTM
0.976	0.9904728	0.9888971	54112605	0.97	LSTM
0.986	0.9906528	0.9891173	54521663	0.98	LSTM
0.996	0.9907173	0.9891941	54716174	0.99	LSTM
0.757	0.7838920	0.7764890	22951433	0.75	LightGBM
0.767	0.7954688	0.7882016	23563522	0.76	LightGBM
0.777	0.8069795	0.7998487	24194941	0.77	LightGBM
0.787	0.8184317	0.8115044	24855923	0.78	LightGBM
0.797	0.8298863	0.8231205	25551814	0.79	LightGBM
0.827	0.8410667	0.8344968	26275084	0.80	LightGBM

0.8111	0.8523301	0.8459967	27044064	0.81	LightGBM
0.828	0.8633996	0.8572803	27842590	0.82	LightGBM
0.837	0.8740500	0.8681867	28681723	0.83	LightGBM
0.847	0.8849539	0.8793887	29588552	0.84	LightGBM
0.857	0.8954805	0.8901875	30542409	0.85	LightGBM
0.867	0.9055309	0.9005203	31556964	0.86	LightGBM
0.877	0.9155343	0.9108468	32652336	0.87	LightGBM
0.887	0.9252367	0.9208841	33827128	0.88	LightGBM
0.897	0.9344947	0.9304836	35100378	0.89	LightGBM
0.927	0.9433928	0.9397506	36491354	0.90	LightGBM
0.9111	0.9519355	0.9486676	38016950	0.91	LightGBM
0.928	0.9600220	0.9571309	39713697	0.92	LightGBM
0.937	0.9675030	0.9649933	41607714	0.93	LightGBM
0.947	0.9743430	0.9722511	43735643	0.94	LightGBM
0.957	0.9804146	0.9787398	46134069	0.95	LightGBM
0.967	0.9853680	0.9840576	48762863	0.96	LightGBM
0.977	0.9888626	0.9878213	51404849	0.97	LightGBM
0.987	0.9906184	0.9897464	53654741	0.98	LightGBM
0.997	0.9911204	0.9903028	55007180	0.99	LightGBM
0.758	0.7886884	0.7804357	23165942	0.75	RF
0.768	0.8002095	0.7920976	23780559	0.76	RF
0.778	0.8117368	0.8037632	24423236	0.77	RF
0.788	0.8227839	0.8150220	25080150	0.78	RF
0.798	0.8339637	0.8264032	25777122	0.79	RF
0.829	0.8452464	0.8379143	26511664	0.80	RF
0.8112	0.8562051	0.8491038	27270631	0.81	RF
0.8210	0.8668791	0.8600350	28074305	0.82	RF
0.838	0.8774393	0.8709093	28923302	0.83	RF
0.848	0.8880712	0.8818368	29826105	0.84	RF
0.858	0.8982209	0.8923018	30779079	0.85	RF
0.868	0.9082311	0.9026407	31797326	0.86	RF
0.878	0.9181172	0.9128968	32897234	0.87	RF
0.888	0.9274340	0.9225978	34063687	0.88	RF

0.898	0.9365942	0.9321330	35336800	0.89	RF
0.929	0.9452740	0.9412306	36725333	0.90	RF
0.9112	0.9535515	0.9499226	38240774	0.91	RF
0.9210	0.9613488	0.9581522	39926043	0.92	RF
0.938	0.9685944	0.9658514	41797421	0.93	RF
0.948	0.9751828	0.9729113	43898464	0.94	RF
0.958	0.9809633	0.9791664	46243299	0.95	RF
0.968	0.9856789	0.9842950	48785247	0.96	RF
0.978	0.9889435	0.9878695	51345735	0.97	RF
0.988	0.9906379	0.9897526	53557385	0.98	RF
0.998	0.9911527	0.9903374	54972161	0.99	RF

Table 16: IPM values for MAN

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.6734819	0.7103484	91837.78	0.75	Croston
0.76	0.6822701	0.7220733	94884.23	0.76	Croston
0.77	0.6897269	0.7339199	98059.45	0.77	Croston
0.78	0.6972481	0.7456895	101365.15	0.78	Croston
0.79	0.7054504	0.7574242	104914.72	0.79	Croston
0.8	0.7138291	0.7691326	108576.71	0.80	Croston
0.81	0.7217973	0.7809129	112392.51	0.81	Croston
0.82	0.7305667	0.7926313	116337.22	0.82	Croston
0.83	0.7384567	0.8045922	120534.61	0.83	Croston
0.84	0.7461041	0.8167415	124848.36	0.84	Croston
0.85	0.7536951	0.8289883	129343.62	0.85	Croston
0.86	0.7606159	0.8414460	133875.53	0.86	Croston
0.87	0.7674174	0.8540950	138803.38	0.87	Croston
0.88	0.7745193	0.8672375	143943.98	0.88	Croston
0.89	0.7808530	0.8797479	149199.60	0.89	Croston
0.9	0.7867527	0.8892916	154504.86	0.90	Croston
0.91	0.7923890	0.8974688	160017.99	0.91	Croston
0.92	0.7976095	0.9053799	165952.14	0.92	Croston

0.93	0.8029400	0.9133017	171792.17	0.93	Croston
0.94	0.8074070	0.9211997	177939.54	0.94	Croston
0.95	0.8118564	0.9292473	184177.72	0.95	Croston
0.96	0.8154405	0.9371947	190114.32	0.96	Croston
0.97	0.8188170	0.9447769	196038.71	0.97	Croston
0.98	0.8212882	0.9519788	202035.89	0.98	Croston
0.99	0.8230571	0.9575141	207498.77	0.99	Croston
0.751	0.6876678	0.7156073	98888.94	0.75	SBA
0.761	0.6952528	0.7272181	101986.33	0.76	SBA
0.771	0.7030550	0.7387451	105212.70	0.77	SBA
0.781	0.7101538	0.7504206	108470.95	0.78	SBA
0.791	0.7177245	0.7621423	111918.11	0.79	SBA
0.810	0.7250117	0.7741414	115430.39	0.80	SBA
0.811	0.7328877	0.7858138	119203.82	0.81	SBA
0.821	0.7404670	0.7976427	122979.45	0.82	SBA
0.831	0.7478598	0.8096890	127075.19	0.83	SBA
0.841	0.7544639	0.8216806	131322.18	0.84	SBA
0.851	0.7613051	0.8338221	135743.79	0.85	SBA
0.861	0.7680029	0.8462160	140223.81	0.86	SBA
0.871	0.7740193	0.8588503	145014.96	0.87	SBA
0.881	0.7801505	0.8718868	149867.37	0.88	SBA
0.891	0.7857263	0.8836114	154859.85	0.89	SBA
0.910	0.7910352	0.8923155	159856.94	0.90	SBA
0.911	0.7960007	0.9001410	164990.74	0.91	SBA
0.921	0.8009015	0.9078614	170427.01	0.92	SBA
0.931	0.8053932	0.9154758	175796.85	0.93	SBA
0.941	0.8095474	0.9231161	181355.49	0.94	SBA
0.951	0.8133805	0.9308997	186830.82	0.95	SBA
0.961	0.8167299	0.9385184	191978.21	0.96	SBA
0.971	0.8195780	0.9458860	197289.51	0.97	SBA
0.981	0.8217509	0.9529505	202793.45	0.98	SBA
0.991	0.8231120	0.9576565	208014.79	0.99	SBA
0.752	0.6545046	0.6905383	84858.02	0.75	DLP

0.762	0.6630815	0.7021365	87695.83	0.76	DLP
0.772	0.6717917	0.7138837	90569.16	0.77	DLP
0.782	0.6799123	0.7257599	93660.23	0.78	DLP
0.792	0.6883098	0.7376243	96808.77	0.79	DLP
0.812	0.6959622	0.7494419	100092.04	0.80	DLP
0.813	0.7047495	0.7614159	103632.11	0.81	DLP
0.822	0.7127020	0.7732461	107359.62	0.82	DLP
0.832	0.7208125	0.7851467	111282.29	0.83	DLP
0.842	0.7292435	0.7971698	115339.72	0.84	DLP
0.852	0.7372273	0.8094323	119605.14	0.85	DLP
0.862	0.7455483	0.8216600	124146.02	0.86	DLP
0.872	0.7536078	0.8342136	128923.78	0.87	DLP
0.882	0.7614003	0.8470201	133949.58	0.88	DLP
0.892	0.7689359	0.8603988	139303.70	0.89	DLP
0.912	0.7764201	0.8741181	144944.51	0.90	DLP
0.913	0.7828239	0.8865018	150818.98	0.91	DLP
0.922	0.7897419	0.8972874	156895.23	0.92	DLP
0.932	0.7955845	0.9060207	163353.24	0.93	DLP
0.942	0.8016658	0.9148797	170170.83	0.94	DLP
0.952	0.8072391	0.9238029	177190.94	0.95	DLP
0.962	0.8122260	0.9327367	184229.80	0.96	DLP
0.972	0.8164340	0.9413075	191508.20	0.97	DLP
0.982	0.8201018	0.9499886	198947.83	0.98	DLP
0.992	0.8226713	0.9570829	205993.28	0.99	DLP
0.753	0.7317528	0.7429168	110022.12	0.75	Willemain
0.763	0.7380484	0.7521084	112695.86	0.76	Willemain
0.773	0.7431109	0.7613291	115480.93	0.77	Willemain
0.783	0.7483886	0.7703278	118207.01	0.78	Willemain
0.793	0.7532631	0.7792921	121118.09	0.79	Willemain
0.814	0.7585389	0.7882145	124210.93	0.80	Willemain
0.815	0.7637725	0.7970473	127283.32	0.81	Willemain
0.823	0.7688027	0.8056975	130431.12	0.82	Willemain
0.833	0.7735231	0.8142114	133848.77	0.83	Willemain

0.843	0.7780662	0.8226621	137273.41	0.84	Willemain
0.853	0.7829727	0.8309915	140920.21	0.85	Willemain
0.863	0.7875419	0.8391601	144816.29	0.86	Willemain
0.873	0.7921167	0.8472863	148811.68	0.87	Willemain
0.883	0.7967370	0.8555028	153085.46	0.88	Willemain
0.893	0.8010899	0.8638021	157640.41	0.89	Willemain
0.914	0.8056166	0.8722287	162492.77	0.90	Willemain
0.915	0.8098164	0.8805784	167622.13	0.91	Willemain
0.923	0.8139718	0.8888368	173153.51	0.92	Willemain
0.933	0.8185462	0.8973222	179150.53	0.93	Willemain
0.943	0.8229966	0.9055288	185680.07	0.94	Willemain
0.953	0.8275375	0.9138365	192784.01	0.95	Willemain
0.963	0.8318687	0.9225498	201075.96	0.96	Willemain
0.973	0.8361934	0.9309693	210592.45	0.97	Willemain
0.983	0.8409132	0.9396269	222478.75	0.98	Willemain
0.993	0.8462830	0.9497288	239581.73	0.99	Willemain
0.754	0.0721134	0.3335347	22396.94	0.75	QR
0.764	0.0724512	0.3352854	22767.45	0.76	QR
0.774	0.0727295	0.3369178	23146.95	0.77	QR
0.784	0.0730660	0.3385574	23535.40	0.78	QR
0.794	0.0733426	0.3400752	23926.25	0.79	QR
0.816	0.0736226	0.3415268	24303.00	0.80	QR
0.817	0.0738875	0.3429913	24667.37	0.81	QR
0.824	0.0741416	0.3444181	25030.53	0.82	QR
0.834	0.0744050	0.3457693	25378.74	0.83	QR
0.844	0.0746576	0.3471447	25723.38	0.84	QR
0.854	0.0749005	0.3485228	26089.59	0.85	QR
0.864	0.0751180	0.3498607	26472.01	0.86	QR
0.874	0.0753065	0.3511124	26861.59	0.87	QR
0.884	0.0755213	0.3523025	27265.27	0.88	QR
0.894	0.0757314	0.3534515	27683.69	0.89	QR
0.916	0.0759017	0.3545510	28109.33	0.90	QR
0.917	0.0760837	0.3556728	28536.72	0.91	QR

0.924	0.0762825	0.3567581	28973.04	0.92	QR
0.934	0.0764867	0.3578672	29425.91	0.93	QR
0.944	0.0766863	0.3590122	29877.16	0.94	QR
0.954	0.0768740	0.3602274	30317.93	0.95	QR
0.964	0.0770062	0.3612065	30744.03	0.96	QR
0.974	0.0771204	0.3619326	31163.56	0.97	QR
0.984	0.0772336	0.3623329	31512.56	0.98	QR
0.994	0.0772992	0.3625415	31769.57	0.99	QR
0.755	0.6596059	0.7897659	80579.72	0.75	MLP
0.765	0.6673476	0.8005812	83297.69	0.76	MLP
0.775	0.6744515	0.8114514	86069.40	0.77	MLP
0.785	0.6808039	0.8221566	89049.64	0.78	MLP
0.795	0.6883513	0.8331298	92157.32	0.79	MLP
0.818	0.6962391	0.8427137	95384.52	0.80	MLP
0.819	0.7025343	0.8519701	98722.47	0.81	MLP
0.825	0.7097596	0.8611979	102347.16	0.82	MLP
0.835	0.7173899	0.8696013	106094.04	0.83	MLP
0.845	0.7241538	0.8767716	109876.73	0.84	MLP
0.855	0.7316017	0.8839304	114007.29	0.85	MLP
0.865	0.7388830	0.8908905	118294.48	0.86	MLP
0.875	0.7453045	0.8975039	122771.30	0.87	MLP
0.885	0.7518215	0.9036608	127572.35	0.88	MLP
0.895	0.7588357	0.9095077	132584.79	0.89	MLP
0.918	0.7649809	0.9152244	137934.24	0.90	MLP
0.919	0.7709382	0.9206013	143664.85	0.91	MLP
0.925	0.7775365	0.9257413	149539.11	0.92	MLP
0.935	0.7830428	0.9308491	155978.70	0.93	MLP
0.945	0.7890872	0.9356755	162704.39	0.94	MLP
0.955	0.7953369	0.9400991	169973.19	0.95	MLP
0.965	0.8011999	0.9441164	177848.16	0.96	MLP
0.975	0.8062980	0.9480344	186373.57	0.97	MLP
0.985	0.8118424	0.9518352	195094.89	0.98	MLP
0.995	0.8172846	0.9552229	204567.78	0.99	MLP

0.756	0.7122287	0.9330413	136490.63	0.75	LSTM
0.766	0.7143208	0.9342090	138255.93	0.76	LSTM
0.776	0.7172173	0.9353486	140036.74	0.77	LSTM
0.786	0.7197988	0.9364373	141857.14	0.78	LSTM
0.796	0.7223669	0.9375012	143571.26	0.79	LSTM
0.820	0.7263994	0.9385176	145368.65	0.80	LSTM
0.8110	0.7287696	0.9394547	147308.27	0.81	LSTM
0.826	0.7314167	0.9403530	149113.81	0.82	LSTM
0.836	0.7345666	0.9412972	151009.33	0.83	LSTM
0.846	0.7372089	0.9422051	152824.33	0.84	LSTM
0.856	0.7399918	0.9431360	154796.77	0.85	LSTM
0.866	0.7428663	0.9440291	156690.13	0.86	LSTM
0.876	0.7452654	0.9449482	158674.83	0.87	LSTM
0.886	0.7476752	0.9458216	160637.67	0.88	LSTM
0.896	0.7510076	0.9466781	162764.49	0.89	LSTM
0.920	0.7539560	0.9475199	164864.80	0.90	LSTM
0.9110	0.7577999	0.9483612	167048.34	0.91	LSTM
0.926	0.7603664	0.9490668	169132.40	0.92	LSTM
0.936	0.7631713	0.9497692	171421.89	0.93	LSTM
0.946	0.7661285	0.9504722	173912.50	0.94	LSTM
0.956	0.7698157	0.9511740	176522.81	0.95	LSTM
0.966	0.7730635	0.9517907	179303.93	0.96	LSTM
0.976	0.7771721	0.9524930	182384.24	0.97	LSTM
0.986	0.7821269	0.9533309	185928.12	0.98	LSTM
0.996	0.7883397	0.9544392	190842.58	0.99	LSTM
0.757	0.7180564	0.8531650	120812.44	0.75	LightGBM
0.767	0.7237552	0.8585625	123228.90	0.76	LightGBM
0.777	0.7285607	0.8640129	125843.17	0.77	LightGBM
0.787	0.7339752	0.8694848	128499.13	0.78	LightGBM
0.797	0.7394325	0.8749275	131190.34	0.79	LightGBM
0.827	0.7444716	0.8800498	133916.31	0.80	LightGBM
0.8111	0.7494324	0.8847907	136827.46	0.81	LightGBM
0.828	0.7539437	0.8893573	139578.73	0.82	LightGBM

0.837	0.7581147	0.8936285	142445.86	0.83	LightGBM
0.847	0.7625703	0.8976344	145371.04	0.84	LightGBM
0.857	0.7666324	0.9016764	148468.00	0.85	LightGBM
0.867	0.7718215	0.9057431	151610.73	0.86	LightGBM
0.877	0.7757529	0.9098158	154814.41	0.87	LightGBM
0.887	0.7796255	0.9136035	158313.27	0.88	LightGBM
0.897	0.7835963	0.9171662	161732.58	0.89	LightGBM
0.927	0.7876064	0.9207317	165441.84	0.90	LightGBM
0.9111	0.7913682	0.9240437	169125.12	0.91	LightGBM
0.928	0.7954508	0.9274640	173091.95	0.92	LightGBM
0.937	0.7992033	0.9308061	177254.67	0.93	LightGBM
0.947	0.8025041	0.9342717	181603.16	0.94	LightGBM
0.957	0.8058832	0.9379568	186018.03	0.95	LightGBM
0.967	0.8092889	0.9415628	190362.92	0.96	LightGBM
0.977	0.8125164	0.9451398	195094.65	0.97	LightGBM
0.987	0.8156188	0.9484138	200405.96	0.98	LightGBM
0.997	0.8188207	0.9530141	205553.88	0.99	LightGBM
0.758	0.7158143	0.8351692	111821.03	0.75	RF
0.768	0.7221260	0.8417242	114513.47	0.76	RF
0.778	0.7275667	0.8481153	117324.05	0.77	RF
0.788	0.7330107	0.8543450	120203.21	0.78	RF
0.798	0.7393551	0.8604996	123190.22	0.79	RF
0.829	0.7440802	0.8666099	126191.57	0.80	RF
0.8112	0.7496285	0.8727799	129317.11	0.81	RF
0.8210	0.7547958	0.8788146	132373.82	0.82	RF
0.838	0.7596620	0.8841959	135606.39	0.83	RF
0.848	0.7643886	0.8894140	138847.19	0.84	RF
0.858	0.7687151	0.8945088	142269.27	0.85	RF
0.868	0.7741417	0.8996247	145873.25	0.86	RF
0.878	0.7781993	0.9044567	149495.70	0.87	RF
0.888	0.7822405	0.9093267	153391.79	0.88	RF
0.898	0.7862213	0.9142546	157220.34	0.89	RF
0.929	0.7896940	0.9188878	161362.01	0.90	RF

0.9112	0.7936245	0.9233114	165487.44	0.91	RF
0.9210	0.7974709	0.9275943	169890.82	0.92	RF
0.938	0.8014801	0.9319685	174533.68	0.93	RF
0.948	0.8046493	0.9361680	179508.28	0.94	RF
0.958	0.8079550	0.9404348	184571.31	0.95	RF
0.968	0.8113828	0.9445644	189544.48	0.96	RF
0.978	0.8142065	0.9482306	194880.82	0.97	RF
0.988	0.8172539	0.9516104	200796.96	0.98	RF
0.998	0.8202057	0.9553530	206611.10	0.99	RF

Table 17: IPM values for BRAF

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.8471975	0.6616482	470776.5	0.75	Croston
0.76	0.8540241	0.6715195	486981.0	0.76	Croston
0.77	0.8610106	0.6816066	502468.9	0.77	Croston
0.78	0.8669248	0.6911707	517899.5	0.78	Croston
0.79	0.8728567	0.7006825	535225.0	0.79	Croston
0.8	0.8785274	0.7101158	550467.6	0.80	Croston
0.81	0.8837557	0.7196015	567575.8	0.81	Croston
0.82	0.8884491	0.7289956	585668.2	0.82	Croston
0.83	0.8930882	0.7385728	602292.0	0.83	Croston
0.84	0.8977482	0.7480519	621434.5	0.84	Croston
0.85	0.9025882	0.7576160	638283.1	0.85	Croston
0.86	0.9067809	0.7667944	656148.7	0.86	Croston
0.87	0.9111361	0.7759989	675870.8	0.87	Croston
0.88	0.9146816	0.7848897	695101.6	0.88	Croston
0.89	0.9182887	0.7932378	716236.0	0.89	Croston
0.9	0.9212318	0.8010826	733912.7	0.90	Croston
0.91	0.9239086	0.8082344	751782.1	0.91	Croston
0.92	0.9260444	0.8144971	767668.0	0.92	Croston
0.93	0.9277455	0.8200016	780173.3	0.93	Croston
0.94	0.9289331	0.8243554	789283.0	0.94	Croston

0.95	0.9298184	0.8272580	795923.5	0.95	Croston
0.96	0.9303918	0.8291865	799556.5	0.96	Croston
0.97	0.9307554	0.8304613	802117.8	0.97	Croston
0.98	0.9309457	0.8314353	803688.3	0.98	Croston
0.99	0.9310939	0.8317491	804622.7	0.99	Croston
0.751	0.8621064	0.6871960	514959.9	0.75	SBA
0.761	0.8681459	0.6962567	531494.3	0.76	SBA
0.771	0.8735589	0.7050952	546324.7	0.77	SBA
0.781	0.8788641	0.7139794	561180.2	0.78	SBA
0.791	0.8837524	0.7225433	575057.2	0.79	SBA
0.810	0.8884374	0.7306757	588804.4	0.80	SBA
0.811	0.8928821	0.7389389	603120.1	0.81	SBA
0.821	0.8969631	0.7470386	619403.4	0.82	SBA
0.831	0.9008089	0.7550010	635888.0	0.83	SBA
0.841	0.9046769	0.7630746	651059.6	0.84	SBA
0.851	0.9080328	0.7711351	666516.4	0.85	SBA
0.861	0.9114622	0.7792545	681360.5	0.86	SBA
0.871	0.9148705	0.7872562	698796.8	0.87	SBA
0.881	0.9180439	0.7946956	717011.4	0.88	SBA
0.891	0.9208300	0.8017102	733455.9	0.89	SBA
0.910	0.9234552	0.8083063	749979.0	0.90	SBA
0.911	0.9257096	0.8141899	765339.6	0.91	SBA
0.921	0.9271480	0.8190537	777979.1	0.92	SBA
0.931	0.9284117	0.8232702	786720.0	0.93	SBA
0.941	0.9294110	0.8265258	794404.0	0.94	SBA
0.951	0.9299147	0.8283170	798520.8	0.95	SBA
0.961	0.9304999	0.8297422	800921.4	0.96	SBA
0.971	0.9307714	0.8306378	802626.7	0.97	SBA
0.981	0.9309417	0.8311477	803764.2	0.98	SBA
0.991	0.9310831	0.8316053	804599.3	0.99	SBA
0.752	0.8192422	0.6204566	417902.0	0.75	DLP
0.762	0.8255295	0.6309490	431207.7	0.76	DLP
0.772	0.8327681	0.6414936	445646.5	0.77	DLP

0.782	0.8399698	0.6519860	460692.0	0.78	DLP
0.792	0.8468959	0.6624457	475475.4	0.79	DLP
0.812	0.8526609	0.6720883	491744.4	0.80	DLP
0.813	0.8585983	0.6819988	507259.8	0.81	DLP
0.822	0.8660542	0.6921578	523995.5	0.82	DLP
0.832	0.8721237	0.7021534	540676.8	0.83	DLP
0.842	0.8783183	0.7122339	558329.4	0.84	DLP
0.852	0.8837182	0.7221249	576553.4	0.85	DLP
0.862	0.8894026	0.7324015	595294.2	0.86	DLP
0.872	0.8948346	0.7427567	615321.0	0.87	DLP
0.882	0.8999352	0.7533406	635709.7	0.88	DLP
0.892	0.9048567	0.7639049	654929.9	0.89	DLP
0.912	0.9095700	0.7743253	676322.1	0.90	DLP
0.913	0.9144306	0.7847654	698498.2	0.91	DLP
0.922	0.9184207	0.7943949	719947.3	0.92	DLP
0.932	0.9217329	0.8030961	741686.4	0.93	DLP
0.942	0.9245726	0.8115619	760940.7	0.94	DLP
0.952	0.9269477	0.8186353	776509.7	0.95	DLP
0.962	0.9287340	0.8243685	787621.5	0.96	DLP
0.972	0.9298922	0.8279117	795257.9	0.97	DLP
0.982	0.9306766	0.8303371	800747.8	0.98	DLP
0.992	0.9310382	0.8316641	803995.3	0.99	DLP
0.753	0.8991754	0.6952827	622395.8	0.75	Willemain
0.763	0.9028343	0.7018396	639500.9	0.76	Willemain
0.773	0.9060143	0.7084554	655122.5	0.77	Willemain
0.783	0.9090813	0.7147966	671197.2	0.78	Willemain
0.793	0.9118932	0.7210920	687383.8	0.79	Willemain
0.814	0.9153302	0.7276555	702066.5	0.80	Willemain
0.815	0.9182581	0.7335717	715781.1	0.81	Willemain
0.823	0.9208133	0.7393703	731683.1	0.82	Willemain
0.833	0.9235740	0.7453520	746470.9	0.83	Willemain
0.843	0.9263472	0.7513467	761777.5	0.84	Willemain
0.853	0.9290112	0.7572630	779074.2	0.85	Willemain

0.863	0.9315929	0.7630812	795717.8	0.86	Willemain
0.873	0.9343491	0.7691020	815941.1	0.87	Willemain
0.883	0.9371305	0.7751294	835074.0	0.88	Willemain
0.893	0.9398736	0.7813595	857529.6	0.89	Willemain
0.914	0.9424879	0.7873934	881211.2	0.90	Willemain
0.915	0.9451344	0.7935385	906691.5	0.91	Willemain
0.923	0.9476713	0.7998601	934402.0	0.92	Willemain
0.933	0.9501737	0.8062405	964861.9	0.93	Willemain
0.943	0.9529329	0.8128694	1000503.4	0.94	Willemain
0.953	0.9554920	0.8193805	1039882.4	0.95	Willemain
0.963	0.9584246	0.8262251	1085549.6	0.96	Willemain
0.973	0.9613552	0.8335273	1137736.3	0.97	Willemain
0.983	0.9649009	0.8418950	1208539.0	0.98	Willemain
0.993	0.9697344	0.8526162	1328027.6	0.99	Willemain
0.754	0.0000000	0.0000000	0.0	0.75	QR
0.764	0.0000000	0.0000000	0.0	0.76	QR
0.774	0.0000000	0.0000000	0.0	0.77	QR
0.784	0.0000000	0.0000000	0.0	0.78	QR
0.794	0.0000000	0.0000000	0.0	0.79	QR
0.816	0.0000000	0.0000000	0.0	0.80	QR
0.817	0.0000000	0.0000000	0.0	0.81	QR
0.824	0.0000000	0.0000000	0.0	0.82	QR
0.834	0.0000000	0.0000000	0.0	0.83	QR
0.844	0.0000000	0.0000000	0.0	0.84	QR
0.854	0.0000000	0.0000000	0.0	0.85	QR
0.864	0.0000000	0.0000000	0.0	0.86	QR
0.874	0.0000000	0.0000000	0.0	0.87	QR
0.884	0.0000000	0.0000000	0.0	0.88	QR
0.894	0.0000000	0.0000000	0.0	0.89	QR
0.916	0.0000000	0.0000000	0.0	0.90	QR
0.917	0.0000000	0.0000000	0.0	0.91	QR
0.924	0.0000000	0.0000000	0.0	0.92	QR
0.934	0.0000000	0.0000000	0.0	0.93	QR

0.944	0.0000000	0.0000000	0.0	0.94	QR
0.954	0.0000000	0.0000000	0.0	0.95	QR
0.964	0.0000000	0.0000000	0.0	0.96	QR
0.974	0.0000000	0.0000000	0.0	0.97	QR
0.984	0.0000000	0.0000000	0.0	0.98	QR
0.994	0.0000000	0.0000000	0.0	0.99	QR
0.755	0.8340548	0.6183123	430808.5	0.75	MLP
0.765	0.8393845	0.6275822	442888.1	0.76	MLP
0.775	0.8452438	0.6366430	454952.8	0.77	MLP
0.785	0.8503244	0.6457037	466379.5	0.78	MLP
0.795	0.8552963	0.6547382	479267.4	0.79	MLP
0.818	0.8606471	0.6637728	492356.6	0.80	MLP
0.819	0.8652614	0.6728531	504102.1	0.81	MLP
0.825	0.8699562	0.6820708	517406.5	0.82	MLP
0.835	0.8746802	0.6910073	532508.7	0.83	MLP
0.845	0.8791545	0.6999830	547615.0	0.84	MLP
0.855	0.8836286	0.7087365	562361.6	0.85	MLP
0.865	0.8875489	0.7171042	578805.1	0.86	MLP
0.875	0.8917750	0.7252497	594412.2	0.87	MLP
0.885	0.8957153	0.7334737	611318.2	0.88	MLP
0.895	0.8996030	0.7417172	629772.7	0.89	MLP
0.918	0.9032226	0.7499935	646974.6	0.90	MLP
0.919	0.9065116	0.7582697	664904.5	0.91	MLP
0.925	0.9099063	0.7668728	683856.8	0.92	MLP
0.935	0.9135182	0.7750706	704896.3	0.93	MLP
0.945	0.9166000	0.7831246	726250.5	0.94	MLP
0.955	0.9195204	0.7913877	748171.1	0.95	MLP
0.965	0.9222596	0.7991410	769996.9	0.96	MLP
0.975	0.9248158	0.8067243	785247.0	0.97	MLP
0.985	0.9271420	0.8145298	797653.5	0.98	MLP
0.995	0.9293632	0.8230545	803743.8	0.99	MLP
0.756	0.7014544	0.5250042	272641.5	0.75	LSTM
0.766	0.7060721	0.5321086	277617.7	0.76	LSTM

0.776	0.7108215	0.5395830	282375.9	0.77	LSTM
0.786	0.7157683	0.5468631	287644.0	0.78	LSTM
0.796	0.7209316	0.5546428	292715.3	0.79	LSTM
0.820	0.7253032	0.5622283	300824.4	0.80	LSTM
0.8110	0.7306305	0.5702670	308732.3	0.81	LSTM
0.826	0.7363606	0.5787219	317198.9	0.82	LSTM
0.836	0.7415890	0.5872232	325660.0	0.83	LSTM
0.846	0.7470457	0.5956782	335989.7	0.84	LSTM
0.856	0.7528613	0.6050397	347384.1	0.85	LSTM
0.866	0.7588119	0.6145770	357357.2	0.86	LSTM
0.876	0.7641181	0.6240218	369353.3	0.87	LSTM
0.886	0.7700185	0.6340123	384343.4	0.88	LSTM
0.896	0.7771581	0.6447152	397413.3	0.89	LSTM
0.920	0.7834761	0.6552515	410581.0	0.90	LSTM
0.9110	0.7894369	0.6667314	423831.5	0.91	LSTM
0.926	0.7953066	0.6783963	439116.2	0.92	LSTM
0.936	0.8016909	0.6910973	453652.7	0.93	LSTM
0.946	0.8081744	0.7045938	468567.8	0.94	LSTM
0.956	0.8147979	0.7188951	488337.5	0.95	LSTM
0.966	0.8221416	0.7350003	509763.4	0.96	LSTM
0.976	0.8295524	0.7527705	539832.2	0.97	LSTM
0.986	0.8377555	0.7735472	578794.3	0.98	LSTM
0.996	0.8481711	0.8006882	636350.2	0.99	LSTM
0.757	0.8511247	0.6815935	569353.2	0.75	LightGBM
0.767	0.8560846	0.6882943	579225.1	0.76	LightGBM
0.777	0.8604646	0.6946943	589185.0	0.77	LightGBM
0.787	0.8642713	0.7009897	600279.9	0.78	LightGBM
0.797	0.8691219	0.7072721	610830.0	0.79	LightGBM
0.827	0.8729240	0.7134564	619986.2	0.80	LightGBM
0.8111	0.8766773	0.7195361	629391.8	0.81	LightGBM
0.828	0.8806531	0.7256354	641045.7	0.82	LightGBM
0.837	0.8845927	0.7320093	650425.3	0.83	LightGBM
0.847	0.8879657	0.7379779	660086.3	0.84	LightGBM

0.857	0.8919683	0.7443191	671623.8	0.85	LightGBM
0.867	0.8955900	0.7501634	681238.7	0.86	LightGBM
0.877	0.8991234	0.7562693	691648.9	0.87	LightGBM
0.887	0.9022859	0.7622575	702915.4	0.88	LightGBM
0.897	0.9056808	0.7683764	712681.7	0.89	LightGBM
0.927	0.9086366	0.7742469	722781.5	0.90	LightGBM
0.9111	0.9114513	0.7801697	732693.4	0.91	LightGBM
0.928	0.9143668	0.7862625	744006.8	0.92	LightGBM
0.937	0.9170591	0.7926560	753825.9	0.93	LightGBM
0.947	0.9200559	0.7991148	764449.8	0.94	LightGBM
0.957	0.9222326	0.8053057	774016.1	0.95	LightGBM
0.967	0.9244816	0.8111827	782993.6	0.96	LightGBM
0.977	0.9263943	0.8164453	791033.0	0.97	LightGBM
0.987	0.9281549	0.8214071	797411.1	0.98	LightGBM
0.997	0.9296302	0.8263885	801248.6	0.99	LightGBM
0.758	0.8494212	0.6743763	564363.6	0.75	RF
0.768	0.8543145	0.6815870	574749.3	0.76	RF
0.778	0.8587587	0.6884316	585053.9	0.77	RF
0.788	0.8626595	0.6952761	595976.0	0.78	RF
0.798	0.8675978	0.7020684	607023.4	0.79	RF
0.829	0.8718695	0.7087953	619049.6	0.80	RF
0.8112	0.8757259	0.7152999	628375.7	0.81	RF
0.8210	0.8801613	0.7219092	638949.4	0.82	RF
0.838	0.8837699	0.7285445	649675.1	0.83	RF
0.848	0.8873996	0.7350818	659623.6	0.84	RF
0.858	0.8914450	0.7414230	671387.5	0.85	RF
0.868	0.8952315	0.7476204	682297.1	0.86	RF
0.878	0.8987417	0.7539093	693541.9	0.87	RF
0.888	0.9018785	0.7600479	704039.9	0.88	RF
0.898	0.9052987	0.7664740	716255.1	0.89	RF
0.929	0.9083204	0.7725733	726330.8	0.90	RF
0.9112	0.9111548	0.7786988	737575.0	0.91	RF
0.9210	0.9142481	0.7850465	747435.2	0.92	RF

0.938	0.9170721	0.7915577	758275.8	0.93	RF
0.948	0.9199740	0.7981735	768725.9	0.94	RF
0.958	0.9224352	0.8047696	778435.6	0.95	RF
0.968	0.9247269	0.8112546	787174.1	0.96	RF
0.978	0.9267072	0.8169552	795051.8	0.97	RF
0.988	0.9285555	0.8221720	800841.5	0.98	RF
0.998	0.9300283	0.8272449	804267.8	0.99	RF

Table 18: IPM values for AUTO

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.8329468	0.7567474	5017531	0.75	Croston
0.76	0.8402146	0.7656285	5142227	0.76	Croston
0.77	0.8469994	0.7743867	5261588	0.77	Croston
0.78	0.8538949	0.7831002	5390169	0.78	Croston
0.79	0.8604689	0.7922047	5526738	0.79	Croston
0.8	0.8680720	0.8017338	5671131	0.80	Croston
0.81	0.8744967	0.8105590	5815014	0.81	Croston
0.82	0.8812242	0.8195295	5967477	0.82	Croston
0.83	0.8879504	0.8286563	6125861	0.83	Croston
0.84	0.8947175	0.8377385	6293616	0.84	Croston
0.85	0.9010144	0.8467537	6470030	0.85	Croston
0.86	0.9073232	0.8555901	6656612	0.86	Croston
0.87	0.9141551	0.8646834	6875842	0.87	Croston
0.88	0.9209396	0.8734975	7116128	0.88	Croston
0.89	0.9273134	0.8821440	7372147	0.89	Croston
0.9	0.9339066	0.8908575	7640692	0.90	Croston
0.91	0.9403568	0.8994035	7928011	0.91	Croston
0.92	0.9463411	0.9081170	8218671	0.92	Croston
0.93	0.9521564	0.9165847	8534195	0.93	Croston
0.94	0.9572339	0.9247844	8872438	0.94	Croston
0.95	0.9618978	0.9328165	9218987	0.95	Croston
0.96	0.9651899	0.9394187	9548346	0.96	Croston

0.97	0.9678386	0.9450713	9834601	0.97	Croston
0.98	0.9691131	0.9487354	10046348	0.98	Croston
0.99	0.9695711	0.9500425	10143738	0.99	Croston
0.751	0.8322482	0.7548930	5045299	0.75	SBA
0.761	0.8391691	0.7633831	5169515	0.76	SBA
0.771	0.8462387	0.7725323	5290591	0.77	SBA
0.781	0.8534080	0.7816145	5419736	0.78	SBA
0.791	0.8598827	0.7906520	5555774	0.79	SBA
0.810	0.8674379	0.8003485	5702517	0.80	SBA
0.811	0.8740309	0.8091514	5853644	0.81	SBA
0.821	0.8807726	0.8183788	6011779	0.82	SBA
0.831	0.8873627	0.8274610	6175794	0.83	SBA
0.841	0.8942052	0.8368336	6346431	0.84	SBA
0.851	0.9011251	0.8462733	6523249	0.85	SBA
0.861	0.9075988	0.8552326	6712056	0.86	SBA
0.871	0.9141817	0.8642812	6930746	0.87	SBA
0.881	0.9213226	0.8734640	7178477	0.88	SBA
0.891	0.9277695	0.8824233	7435141	0.89	SBA
0.910	0.9347111	0.8912038	7701889	0.90	SBA
0.911	0.9404835	0.8993588	7986823	0.91	SBA
0.921	0.9466225	0.9081170	8275807	0.92	SBA
0.931	0.9523630	0.9166183	8576523	0.93	SBA
0.941	0.9572534	0.9248961	8913196	0.94	SBA
0.951	0.9618776	0.9327718	9254337	0.95	SBA
0.961	0.9651897	0.9394522	9578012	0.96	SBA
0.971	0.9678856	0.9450824	9852536	0.97	SBA
0.981	0.9691002	0.9487578	10054906	0.98	SBA
0.991	0.9695612	0.9500313	10145132	0.99	SBA
0.752	0.8649218	0.8003821	5497289	0.75	DLP
0.762	0.8702702	0.8078221	5606740	0.76	DLP
0.772	0.8752208	0.8146477	5718981	0.77	DLP
0.782	0.8806973	0.8218978	5840353	0.78	DLP
0.792	0.8862010	0.8290138	5960239	0.79	DLP

0.812	0.8916353	0.8362527	6089653	0.80	DLP
0.813	0.8971879	0.8434805	6223845	0.81	DLP
0.822	0.9019882	0.8502837	6372101	0.82	DLP
0.832	0.9072480	0.8574110	6515663	0.83	DLP
0.842	0.9122281	0.8641695	6665292	0.84	DLP
0.852	0.9175141	0.8711627	6830613	0.85	DLP
0.862	0.9220320	0.8775079	7005828	0.86	DLP
0.872	0.9265911	0.8842553	7197311	0.87	DLP
0.882	0.9321959	0.8913825	7404860	0.88	DLP
0.892	0.9372717	0.8981411	7621647	0.89	DLP
0.912	0.9424556	0.9050896	7860596	0.90	DLP
0.913	0.9474648	0.9118928	8101133	0.91	DLP
0.922	0.9522477	0.9186961	8362260	0.92	DLP
0.932	0.9563640	0.9250302	8626739	0.93	DLP
0.942	0.9603210	0.9315318	8910057	0.94	DLP
0.952	0.9635259	0.9373296	9220284	0.95	DLP
0.962	0.9662544	0.9423455	9520706	0.96	DLP
0.972	0.9682733	0.9465794	9802891	0.97	DLP
0.982	0.9692229	0.9490482	10029159	0.98	DLP
0.992	0.9695649	0.9500425	10142344	0.99	DLP
0.753	0.8997988	0.8043478	6507536	0.75	Willemain
0.763	0.9046761	0.8117543	6648346	0.76	Willemain
0.773	0.9089963	0.8192055	6777408	0.77	Willemain
0.783	0.9136478	0.8267125	6924428	0.78	Willemain
0.793	0.9183912	0.8340632	7071264	0.79	Willemain
0.814	0.9228423	0.8416931	7226470	0.80	Willemain
0.815	0.9273223	0.8491219	7391799	0.81	Willemain
0.823	0.9315789	0.8561821	7562896	0.82	Willemain
0.833	0.9356001	0.8630524	7742309	0.83	Willemain
0.843	0.9396530	0.8699115	7934413	0.84	Willemain
0.853	0.9438813	0.8768823	8148849	0.85	Willemain
0.863	0.9476493	0.8835404	8381433	0.86	Willemain
0.873	0.9516866	0.8900532	8623983	0.87	Willemain

0.883	0.9554712	0.8964208	8888428	0.88	Willemain
0.893	0.9593095	0.9032128	9154793	0.89	Willemain
0.914	0.9625578	0.9090665	9447451	0.90	Willemain
0.915	0.9661503	0.9154229	9756762	0.91	Willemain
0.923	0.9694671	0.9214107	10102405	0.92	Willemain
0.933	0.9727769	0.9277336	10483332	0.93	Willemain
0.943	0.9763759	0.9341123	10914577	0.94	Willemain
0.953	0.9791150	0.9400442	11411652	0.95	Willemain
0.963	0.9825753	0.9467246	11998278	0.96	Willemain
0.973	0.9854293	0.9530028	12718093	0.97	Willemain
0.983	0.9883984	0.9597390	13665124	0.98	Willemain
0.993	0.9915169	0.9671232	15117727	0.99	Willemain
0.754	0.8346318	0.7667903	5337098	0.75	QR
0.764	0.8417716	0.7758948	5468206	0.76	QR
0.774	0.8487719	0.7849323	5596592	0.77	QR
0.784	0.8549240	0.7935229	5724683	0.78	QR
0.794	0.8617057	0.8022700	5870566	0.79	QR
0.816	0.8688939	0.8115756	6022438	0.80	QR
0.817	0.8752983	0.8200545	6167088	0.81	QR
0.824	0.8819292	0.8286898	6320256	0.82	QR
0.834	0.8879238	0.8368895	6486402	0.83	QR
0.844	0.8942306	0.8457036	6662517	0.84	QR
0.854	0.9004423	0.8541155	6849947	0.85	QR
0.864	0.9071200	0.8625609	7050040	0.86	QR
0.874	0.9130055	0.8707605	7269618	0.87	QR
0.884	0.9194659	0.8787144	7514887	0.88	QR
0.894	0.9257017	0.8868470	7777332	0.89	QR
0.916	0.9324382	0.8951137	8051989	0.90	QR
0.917	0.9389568	0.9031570	8331451	0.91	QR
0.924	0.9450336	0.9113231	8616917	0.92	QR
0.934	0.9499436	0.9193217	8892639	0.93	QR
0.944	0.9544732	0.9267282	9172477	0.94	QR
0.954	0.9582142	0.9335538	9443058	0.95	QR

0.964	0.9612713	0.9397761	9694301	0.96	QR
0.974	0.9634214	0.9445574	9892707	0.97	QR
0.984	0.9645276	0.9472050	10035128	0.98	QR
0.994	0.9649058	0.9483333	10097810	0.99	QR
0.755	0.8285125	0.7493074	4926205	0.75	MLP
0.765	0.8352813	0.7579204	5038137	0.76	MLP
0.775	0.8420558	0.7668238	5157319	0.77	MLP
0.785	0.8492772	0.7759618	5272527	0.78	MLP
0.795	0.8560591	0.7850775	5395827	0.79	MLP
0.818	0.8635762	0.7946289	5534668	0.80	MLP
0.819	0.8703331	0.8038451	5668678	0.81	MLP
0.825	0.8766512	0.8124693	5800296	0.82	MLP
0.835	0.8833355	0.8213839	5948483	0.83	MLP
0.845	0.8900700	0.8304661	6105304	0.84	MLP
0.855	0.8969106	0.8397940	6272145	0.85	MLP
0.865	0.9040234	0.8492783	6447856	0.86	MLP
0.875	0.9106329	0.8585728	6645287	0.87	MLP
0.885	0.9174305	0.8675209	6868555	0.88	MLP
0.895	0.9243187	0.8767930	7115250	0.89	MLP
0.918	0.9312525	0.8861544	7361798	0.90	MLP
0.919	0.9374341	0.8949350	7637018	0.91	MLP
0.925	0.9437640	0.9040395	7924077	0.92	MLP
0.935	0.9498108	0.9132669	8223042	0.93	MLP
0.945	0.9552099	0.9217123	8555155	0.94	MLP
0.955	0.9604402	0.9300237	8912369	0.95	MLP
0.965	0.9642735	0.9373073	9276775	0.96	MLP
0.975	0.9673059	0.9434850	9614660	0.97	MLP
0.985	0.9688953	0.9479311	9921584	0.98	MLP
0.995	0.9695436	0.9498972	10124000	0.99	MLP
0.756	0.8679492	0.8583326	7501396	0.75	LSTM
0.766	0.8710693	0.8657450	7647691	0.76	LSTM
0.776	0.8736223	0.8716998	7792168	0.77	LSTM
0.786	0.8761025	0.8775298	7931698	0.78	LSTM

0.796	0.8788513	0.8835679	8080615	0.79	LSTM
0.820	0.8815305	0.8897726	8221219	0.80	LSTM
0.8110	0.8835187	0.8951861	8371307	0.81	LSTM
0.826	0.8859913	0.9011410	8504272	0.82	LSTM
0.836	0.8881150	0.9065129	8626609	0.83	LSTM
0.846	0.8897427	0.9113850	8759613	0.84	LSTM
0.856	0.8915726	0.9160906	8877998	0.85	LSTM
0.866	0.8930758	0.9208795	8998422	0.86	LSTM
0.876	0.8942679	0.9252103	9109457	0.87	LSTM
0.886	0.8954387	0.9291247	9214587	0.88	LSTM
0.896	0.8968987	0.9334138	9314180	0.89	LSTM
0.920	0.8978198	0.9362039	9393952	0.90	LSTM
0.9110	0.8984489	0.9384942	9466190	0.91	LSTM
0.926	0.8989343	0.9408262	9525111	0.92	LSTM
0.936	0.8994593	0.9429499	9575191	0.93	LSTM
0.946	0.8999275	0.9450321	9614197	0.94	LSTM
0.956	0.9002191	0.9468227	9642146	0.95	LSTM
0.966	0.9004565	0.9484051	9660449	0.96	LSTM
0.976	0.9005629	0.9491130	9669797	0.97	LSTM
0.986	0.9006678	0.9495294	9673210	0.98	LSTM
0.996	0.9007012	0.9497793	9673760	0.99	LSTM
0.757	0.8477569	0.7730238	5486176	0.75	LightGBM
0.767	0.8543257	0.7816703	5606074	0.76	LightGBM
0.777	0.8608721	0.7901828	5737258	0.77	LightGBM
0.787	0.8672507	0.7986058	5863893	0.78	LightGBM
0.797	0.8734238	0.8074087	5992837	0.79	LightGBM
0.827	0.8801922	0.8163122	6141567	0.80	LightGBM
0.8111	0.8864774	0.8245342	6280892	0.81	LightGBM
0.828	0.8926887	0.8328902	6430863	0.82	LightGBM
0.837	0.8979918	0.8409446	6578272	0.83	LightGBM
0.847	0.9042134	0.8492448	6749166	0.84	LightGBM
0.857	0.9100817	0.8572769	6919391	0.85	LightGBM
0.867	0.9161829	0.8654542	7107925	0.86	LightGBM

0.877	0.9216126	0.8730506	7312541	0.87	LightGBM
0.887	0.9272984	0.8807923	7531588	0.88	LightGBM
0.897	0.9333640	0.8890143	7760457	0.89	LightGBM
0.927	0.9388086	0.8965883	7996952	0.90	LightGBM
0.9111	0.9445356	0.9045422	8239906	0.91	LightGBM
0.928	0.9492869	0.9118593	8492556	0.92	LightGBM
0.937	0.9542064	0.9194111	8757203	0.93	LightGBM
0.947	0.9583855	0.9262702	9036235	0.94	LightGBM
0.957	0.9617346	0.9323138	9313630	0.95	LightGBM
0.967	0.9644441	0.9375195	9582979	0.96	LightGBM
0.977	0.9665545	0.9416976	9824667	0.97	LightGBM
0.987	0.9675355	0.9445686	10013037	0.98	LightGBM
0.997	0.9678529	0.9458197	10119789	0.99	LightGBM
0.758	0.8387887	0.7602105	5152951	0.75	RF
0.768	0.8455730	0.7689463	5270515	0.76	RF
0.778	0.8521024	0.7774811	5391144	0.77	RF
0.788	0.8585057	0.7862170	5521086	0.78	RF
0.798	0.8655084	0.7953774	5654829	0.79	RF
0.829	0.8721253	0.8044372	5790769	0.80	RF
0.8112	0.8794235	0.8136758	5934812	0.81	RF
0.8210	0.8860854	0.8227468	6089347	0.82	RF
0.838	0.8921158	0.8314826	6248844	0.83	RF
0.848	0.8983048	0.8402520	6411921	0.84	RF
0.858	0.9042241	0.8486974	6584885	0.85	RF
0.868	0.9110942	0.8579695	6781940	0.86	RF
0.878	0.9178662	0.8669288	6986074	0.87	RF
0.888	0.9241278	0.8757205	7208397	0.88	RF
0.898	0.9303696	0.8841660	7452154	0.89	RF
0.929	0.9366721	0.8927566	7705766	0.90	RF
0.9112	0.9423710	0.9011238	7968575	0.91	RF
0.9210	0.9479481	0.9096251	8235470	0.92	RF
0.938	0.9533785	0.9180929	8524044	0.93	RF
0.948	0.9582467	0.9260467	8843035	0.94	RF

0.958	0.9622083	0.9333974	9158445	0.95	RF
0.968	0.9654225	0.9397203	9473731	0.96	RF
0.978	0.9677966	0.9451830	9764130	0.97	RF
0.988	0.9691091	0.9486125	10000036	0.98	RF
0.998	0.9695553	0.9500201	10138258	0.99	RF

Table 19: IPM values for OIL

	AchievedFillRates_Avg	AchievedFillRates_Total	HoldingCosts	TargetFillRates	Method
0.75	0.5275595	0.6454590	1135064	0.75	Croston
0.76	0.5310058	0.6571786	1154793	0.76	Croston
0.77	0.5348048	0.6689847	1177853	0.77	Croston
0.78	0.5383812	0.6809059	1202464	0.78	Croston
0.79	0.5426188	0.6935326	1225758	0.79	Croston
0.8	0.5463268	0.7064185	1253415	0.80	Croston
0.81	0.5501802	0.7193043	1274390	0.81	Croston
0.82	0.5531894	0.7313983	1293636	0.82	Croston
0.83	0.5562027	0.7441402	1318422	0.83	Croston
0.84	0.5587602	0.7568245	1336561	0.84	Croston
0.85	0.5609495	0.7693360	1357612	0.85	Croston
0.86	0.5629412	0.7813004	1381209	0.86	Croston
0.87	0.5654122	0.7936823	1400563	0.87	Croston
0.88	0.5671082	0.8043798	1418491	0.88	Croston
0.89	0.5687680	0.8142421	1435064	0.89	Croston
0.9	0.5704220	0.8224631	1452596	0.90	Croston
0.91	0.5714504	0.8288269	1465738	0.91	Croston
0.92	0.5724101	0.8344132	1475887	0.92	Croston
0.93	0.5730692	0.8390492	1484204	0.93	Croston
0.94	0.5734948	0.8422455	1488217	0.94	Croston
0.95	0.5740180	0.8447218	1490901	0.95	Croston
0.96	0.5742231	0.8464064	1492516	0.96	Croston
0.97	0.5743567	0.8478605	1493839	0.97	Croston
0.98	0.5744586	0.8488971	1494490	0.98	Croston

0.99	0.5744956	0.8497466	1494814	0.99	Croston
0.751	0.5358301	0.6735199	1190956	0.75	SBA
0.761	0.5391739	0.6844333	1208490	0.76	SBA
0.771	0.5426994	0.6955627	1230968	0.77	SBA
0.781	0.5460520	0.7067928	1248836	0.78	SBA
0.791	0.5496494	0.7180373	1267138	0.79	SBA
0.810	0.5527906	0.7297282	1290784	0.80	SBA
0.811	0.5557823	0.7409727	1311256	0.81	SBA
0.821	0.5577590	0.7514254	1327270	0.82	SBA
0.831	0.5596721	0.7623531	1347900	0.83	SBA
0.841	0.5623521	0.7737416	1364692	0.84	SBA
0.851	0.5640702	0.7846262	1380686	0.85	SBA
0.861	0.5655752	0.7950357	1399277	0.86	SBA
0.871	0.5675294	0.8056467	1414034	0.87	SBA
0.881	0.5689725	0.8144293	1432179	0.88	SBA
0.891	0.5703163	0.8225783	1445824	0.89	SBA
0.910	0.5716159	0.8291004	1461167	0.90	SBA
0.911	0.5725355	0.8343700	1471659	0.91	SBA
0.921	0.5733158	0.8388908	1479975	0.92	SBA
0.931	0.5737959	0.8424902	1487170	0.93	SBA
0.941	0.5740505	0.8451682	1490816	0.94	SBA
0.951	0.5742422	0.8473134	1492466	0.95	SBA
0.961	0.5743335	0.8483500	1493315	0.96	SBA
0.971	0.5744039	0.8490987	1494167	0.97	SBA
0.981	0.5744744	0.8497898	1494642	0.98	SBA
0.991	0.5745055	0.8499050	1494881	0.99	SBA
0.752	0.5060459	0.5970399	1033434	0.75	DLP
0.762	0.5097608	0.6087883	1050455	0.76	DLP
0.772	0.5148500	0.6207383	1071972	0.77	DLP
0.782	0.5196013	0.6332066	1090656	0.78	DLP
0.792	0.5240740	0.6453870	1113367	0.79	DLP
0.812	0.5283609	0.6578697	1137486	0.80	DLP
0.813	0.5325751	0.6701797	1159679	0.81	DLP

0.822	0.5363684	0.6831087	1185096	0.82	DLP
0.832	0.5410228	0.6962969	1210431	0.83	DLP
0.842	0.5450863	0.7100466	1237856	0.84	DLP
0.852	0.5483411	0.7237676	1260436	0.85	DLP
0.862	0.5518549	0.7381796	1283125	0.86	DLP
0.872	0.5554293	0.7522460	1306670	0.87	DLP
0.882	0.5587504	0.7669028	1328350	0.88	DLP
0.892	0.5612524	0.7797023	1356213	0.89	DLP
0.912	0.5637416	0.7924009	1381580	0.90	DLP
0.913	0.5661252	0.8035447	1403616	0.91	DLP
0.922	0.5679355	0.8129463	1425016	0.92	DLP
0.932	0.5696437	0.8219448	1446139	0.93	DLP
0.942	0.5713053	0.8305978	1463376	0.94	DLP
0.952	0.5724804	0.8374367	1476590	0.95	DLP
0.962	0.5733180	0.8423174	1484787	0.96	DLP
0.972	0.5740280	0.8457729	1489927	0.97	DLP
0.982	0.5743595	0.8480621	1493153	0.98	DLP
0.992	0.5744911	0.8496746	1494656	0.99	DLP
0.753	0.5695094	0.6995652	1990949	0.75	Willemain
0.763	0.5711844	0.7062601	2062363	0.76	Willemain
0.773	0.5728001	0.7124798	2141054	0.77	Willemain
0.783	0.5745043	0.7189444	2233420	0.78	Willemain
0.793	0.5759739	0.7251785	2313214	0.79	Willemain
0.814	0.5774800	0.7314127	2382820	0.80	Willemain
0.815	0.5788544	0.7376325	2425857	0.81	Willemain
0.823	0.5801903	0.7438522	2469209	0.82	Willemain
0.833	0.5816016	0.7500864	2503826	0.83	Willemain
0.843	0.5825935	0.7562342	2533258	0.84	Willemain
0.853	0.5838410	0.7626123	2560300	0.85	Willemain
0.863	0.5848600	0.7684865	2586812	0.86	Willemain
0.873	0.5859808	0.7751382	2614986	0.87	Willemain
0.883	0.5868562	0.7813004	2643544	0.88	Willemain
0.893	0.5877528	0.7876929	2680381	0.89	Willemain

0.914	0.5889627	0.7947046	2719862	0.90	Willemain
0.915	0.5898520	0.8011403	2771289	0.91	Willemain
0.923	0.5909364	0.8084111	2848378	0.92	Willemain
0.933	0.5919889	0.8157539	2949632	0.93	Willemain
0.943	0.5934005	0.8232982	3077092	0.94	Willemain
0.953	0.5948616	0.8310729	3240600	0.95	Willemain
0.963	0.5960078	0.8388764	3385530	0.96	Willemain
0.973	0.5972291	0.8474286	3521161	0.97	Willemain
0.983	0.5987732	0.8570030	3692460	0.98	Willemain
0.993	0.6006727	0.8687082	4058969	0.99	Willemain
0.754	0.0000000	0.0000000	0	0.75	QR
0.764	0.0000000	0.0000000	0	0.76	QR
0.774	0.0000000	0.0000000	0	0.77	QR
0.784	0.0000000	0.0000000	0	0.78	QR
0.794	0.0000000	0.0000000	0	0.79	QR
0.816	0.0000000	0.0000000	0	0.80	QR
0.817	0.0000000	0.0000000	0	0.81	QR
0.824	0.0000000	0.0000000	0	0.82	QR
0.834	0.0000000	0.0000000	0	0.83	QR
0.844	0.0000000	0.0000000	0	0.84	QR
0.854	0.0000000	0.0000000	0	0.85	QR
0.864	0.0000000	0.0000000	0	0.86	QR
0.874	0.0000000	0.0000000	0	0.87	QR
0.884	0.0000000	0.0000000	0	0.88	QR
0.894	0.0000000	0.0000000	0	0.89	QR
0.916	0.0000000	0.0000000	0	0.90	QR
0.917	0.0000000	0.0000000	0	0.91	QR
0.924	0.0000000	0.0000000	0	0.92	QR
0.934	0.0000000	0.0000000	0	0.93	QR
0.944	0.0000000	0.0000000	0	0.94	QR
0.954	0.0000000	0.0000000	0	0.95	QR
0.964	0.0000000	0.0000000	0	0.96	QR
0.974	0.0000000	0.0000000	0	0.97	QR

0.984	0.0000000	0.0000000	0	0.98	QR
0.994	0.0000000	0.0000000	0	0.99	QR
0.755	0.5243938	0.6260078	1086340	0.75	MLP
0.765	0.5270800	0.6367916	1098817	0.76	MLP
0.775	0.5297409	0.6480218	1114170	0.77	MLP
0.785	0.5325120	0.6591799	1132361	0.78	MLP
0.795	0.5345350	0.6701077	1146169	0.79	MLP
0.818	0.5372010	0.6804020	1163828	0.80	MLP
0.819	0.5408146	0.6910274	1193037	0.81	MLP
0.825	0.5433223	0.7015089	1208358	0.82	MLP
0.835	0.5453660	0.7109105	1223149	0.83	MLP
0.845	0.5508093	0.7206001	1264008	0.84	MLP
0.855	0.5526015	0.7290803	1279465	0.85	MLP
0.865	0.5547101	0.7379924	1300419	0.86	MLP
0.875	0.5564643	0.7464582	1319739	0.87	MLP
0.885	0.5586799	0.7556151	1336518	0.88	MLP
0.895	0.5604206	0.7644984	1356080	0.89	MLP
0.918	0.5621233	0.7732377	1371877	0.90	MLP
0.919	0.5636906	0.7818907	1389392	0.91	MLP
0.925	0.5653343	0.7905437	1409224	0.92	MLP
0.935	0.5672520	0.8000605	1426577	0.93	MLP
0.945	0.5687305	0.8094621	1442132	0.94	MLP
0.955	0.5701816	0.8174240	1458014	0.95	MLP
0.965	0.5714183	0.8254147	1472676	0.96	MLP
0.975	0.5726887	0.8333333	1485428	0.97	MLP
0.985	0.5734488	0.8396251	1493347	0.98	MLP
0.995	0.5741691	0.8458017	1494710	0.99	MLP
0.756	0.4069484	0.6579309	1124362	0.75	LSTM
0.766	0.4088926	0.6670782	1139081	0.76	LSTM
0.776	0.4107675	0.6765903	1155637	0.77	LSTM
0.786	0.4125323	0.6857375	1171157	0.78	LSTM
0.796	0.4142710	0.6950251	1185402	0.79	LSTM
0.820	0.4159029	0.7039479	1197566	0.80	LSTM

0.8110	0.4175925	0.7129549	1210009	0.81	LSTM
0.826	0.4191486	0.7218777	1227410	0.82	LSTM
0.836	0.4206141	0.7311653	1244290	0.83	LSTM
0.846	0.4220060	0.7391902	1259255	0.84	LSTM
0.856	0.4234965	0.7478605	1271488	0.85	LSTM
0.866	0.4245851	0.7561379	1286373	0.86	LSTM
0.876	0.4256530	0.7642470	1301597	0.87	LSTM
0.886	0.4270023	0.7725525	1315599	0.88	LSTM
0.896	0.4280416	0.7800724	1332313	0.89	LSTM
0.920	0.4291665	0.7879851	1345276	0.90	LSTM
0.9110	0.4300726	0.7953646	1359025	0.91	LSTM
0.926	0.4310538	0.8032492	1370033	0.92	LSTM
0.936	0.4318133	0.8103482	1385171	0.93	LSTM
0.946	0.4326713	0.8174752	1398204	0.94	LSTM
0.956	0.4333577	0.8244058	1409489	0.95	LSTM
0.966	0.4339315	0.8306911	1419049	0.96	LSTM
0.976	0.4345156	0.8372289	1428647	0.97	LSTM
0.986	0.4349956	0.8436264	1437550	0.98	LSTM
0.996	0.4354417	0.8498836	1443817	0.99	LSTM
0.757	0.5293782	0.6707268	1135064	0.75	LightGBM
0.767	0.5320538	0.6784583	1152358	0.76	LightGBM
0.777	0.5346203	0.6861898	1166599	0.77	LightGBM
0.787	0.5373750	0.6944828	1182847	0.78	LightGBM
0.797	0.5391412	0.7023439	1197778	0.79	LightGBM
0.827	0.5412786	0.7104066	1210154	0.80	LightGBM
0.8111	0.5441004	0.7187428	1236457	0.81	LightGBM
0.828	0.5469659	0.7257976	1253380	0.82	LightGBM
0.837	0.5488499	0.7327517	1266402	0.83	LightGBM
0.847	0.5514815	0.7396481	1288652	0.84	LightGBM
0.857	0.5553298	0.7466166	1317334	0.85	LightGBM
0.867	0.5573934	0.7534986	1333051	0.86	LightGBM
0.877	0.5588891	0.7600927	1349666	0.87	LightGBM
0.887	0.5606338	0.7673491	1366254	0.88	LightGBM

0.897	0.5623221	0.7746919	1380133	0.89	LightGBM
0.927	0.5636561	0.7818043	1395004	0.90	LightGBM
0.9111	0.5649547	0.7878657	1406315	0.91	LightGBM
0.928	0.5662645	0.7940711	1420582	0.92	LightGBM
0.937	0.5680122	0.8008235	1436679	0.93	LightGBM
0.947	0.5691217	0.8075616	1449765	0.94	LightGBM
0.957	0.5701485	0.8139542	1460675	0.95	LightGBM
0.967	0.5712112	0.8205483	1473220	0.96	LightGBM
0.977	0.5720797	0.8267680	1483729	0.97	LightGBM
0.987	0.5726873	0.8322247	1490747	0.98	LightGBM
0.997	0.5732721	0.8374798	1491926	0.99	LightGBM
0.758	0.5305996	0.6707988	1137213	0.75	RF
0.768	0.5332257	0.6797829	1153541	0.76	RF
0.778	0.5357677	0.6880759	1168642	0.77	RF
0.788	0.5385163	0.6970744	1184763	0.78	RF
0.798	0.5402120	0.7054826	1200016	0.79	RF
0.829	0.5422816	0.7143371	1212310	0.80	RF
0.8112	0.5451794	0.7233212	1238127	0.81	RF
0.8210	0.5481602	0.7311535	1255870	0.82	RF
0.838	0.5499510	0.7386115	1270069	0.83	RF
0.848	0.5525108	0.7462710	1291934	0.84	RF
0.858	0.5565920	0.7541177	1320192	0.85	RF
0.868	0.5583681	0.7617916	1335970	0.86	RF
0.878	0.5598384	0.7692784	1353943	0.87	RF
0.888	0.5617610	0.7772259	1369345	0.88	RF
0.898	0.5632107	0.7839496	1383854	0.89	RF
0.929	0.5645517	0.7905724	1398883	0.90	RF
0.9112	0.5658591	0.7967490	1410684	0.91	RF
0.9210	0.5671886	0.8029688	1424354	0.92	RF
0.938	0.5689629	0.8098364	1440199	0.93	RF
0.948	0.5699889	0.8163874	1453191	0.94	RF
0.958	0.5710713	0.8229383	1464232	0.95	RF
0.968	0.5722262	0.8298347	1476412	0.96	RF

0.978	0.5730789	0.8364288	1486709	0.97	RF
0.988	0.5736612	0.8414680	1493496	0.98	RF
0.998	0.5742407	0.8462912	1494476	0.99	RF