Predicting Future Aircraft Spare Part Purchases by Using Previous Sales Records and Technical Maintenance Documentation

Master Thesis Aerospace Engineering

Mels L. Wittenberg

TU Delft

Predicting Future Aircraft Spare Part Purchases by Using Previous Sales Records and Technical Maintenance Documentation

Master Thesis Aerospace Engineering

by

Mels L. Wittenberg

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended on 09 October 2023.

Student number:	4430581
Project duration:	September 2022 – October 2023
Thesis committee:	Dr. ir. B.F. Santos
	Dr. A. Bombelli
	Dr. ir. JA Pascoe
	Ir. M. Cappitelli
	Mr. O. Tormaehlen

Front picture taken from [1]

TU Delft

Acknowledgements

This thesis marks the end of an incredible journey that started eight years ago when I first set foot in the Faculty of Aerospace Engineering. As a 17-year-old, I was excited to start studying but unaware of the adventures that awaited me. Fortunately, my excitement persisted, and my passion for aviation grew over the years. Reflecting back on the grateful moments in Delft, I would like to express my gratitude to several individuals who played a significant role in this journey.

I would like to thank my daily supervisor at TU Delft, Bruno Santos. Your support, patience, and insights were indispensable in completing this thesis. Despite your busy schedule, you were always available to meet and listen to the challenges I encountered, providing valuable advice. I also enjoyed our fun conversations, which created a pleasant atmosphere.

I would like to thank Mario Cappitelli and Oliver Tormaehlen from the aftermarket distributor for giving me the opportunity to work on this project. Initially, we had only vague ideas about the project's direction, but over time, we developed a strong research focus together, which we believe will be valuable to the company. I appreciate your enthusiasm, expertise, and the freedom you gave me during this thesis. It is unique that you trusted me with this project in the Netherlands while you are both located abroad. Thank you and I hope to stay in touch.

I would like to express my gratitude to all my friends I have made in the last eight years. Without my study colleagues at NB 2.56, completing this project would have been considerably more challenging and certainly not as enjoyable. I want to extend a special thank you to my colleagues at the faculty's service desk, communication department, and FM. Thanks to your warmth and support, I never felt like just one of the thousands of students, but truly a part of the AE family. In particular during the last few months, when I felt stressed and somewhat uncertain about my progress, your motivation and willingness to chat were always there when I needed it. I will miss you all and I hope this is not a final goodbye.

A special thanks to Femke, who has been with me throughout this entire process, making even the most stressful days more enjoyable. Lastly, I would like to thank my parents. Without them, I would not be where I am today. Throughout my studies, you have been extremely supportive and made me believe that everything was possible! Encourage me to follow my own path and do what I think should be best. Thank you for everything!

Mels L. Wittenberg
Delft, October 2023

Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

AHP	Analytical Hierarchic Procedure
AMM	Aircraft Maintenance Manual
ANN	Artificial Neural Network
AOG	Aircraft on Ground
AR	Auto Regressive
ARIMA	Auto Regressive Integrated Moving Average
ARM	Association Rule Mining
ARMA	Auto Regressive Moving Average
ATT	Air Transportation Association
BP	Back Propagation
CD	Calendar Days
CM	Condition Monitoring
DES	Double Exponential Smoothing
ESS	Essentiality code
FC	Flight Cycles
FH	Flight Hours
FN	False Negative
FP	False Positive
FRNN	Fully Recurrent Neural Network
GA	Genetic Algorithm
GMAMAD/A	Geometric Mean of Mean Absolute Deviation Average
GRNN	Generalized Regression Neural Network
GRU	Gated Recurrent Unit
HT	Hard Time
HW	Holt-Winter model
LSTM	Long-Short Term Memory
MA	Moving Average
MAPE	Mean Absolute Percentage Error
MCC	Matthew Correlation Coefficient

ME	Mean Error
ME	Model Effectiveness
ML	Machine Learning
MLP	Multi-Layer Perceptron
MMEL	Master Minimum Equipment List
MPD	Maintenance Planning Document
MRBR	Maintenance Review Board Report
MRO	Maintenance Repair and Overhaul
MSE	Mean Square Error
MSG	Maintenance Steering Group
OC	On-Condition
OEM	Original Equipment Manufacturer
PMA	Part Manufacturer Approval
PMP	Primary Maintenance Processes
PN	Spare Part
RFM	Recency, Frequency, Monetary
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SBA	Syntetos-Boylan-Approximation
SES	Single Exponential Smoothing
SOFM	Self-Organising Feature Map
SPSO-CM	Proposed Subsequent PN Purchase Order Occurrence Classification Model
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
TES	Triple Exponential Smoothing
TN	True Negative
TP	True Positive
TSB	Tuenter-Syntetos-Babai

List of Symbols

α	Smoothing constant
δ	Seasonality smoothing constant
γ	Trend smoothing constant
$\hat{p}_t$	Inter-demand period at time t
$\hat{z}_t$	Demand size at period t

$\overline{Qnt}_{PN}$	Average quantity for every PN per transaction
AUC_{PR}	Area under the curve for the precision-recall graph
C_{PN}	Unique cluster ID
I_t	Smoothing seasonality index
$N_{PN-customers}$	Number of unique customer purchasing the specified PN
$N_{PN-inter}$	Number of interchangeables for the specified PN
S_t	Predicted demand value at timestamp t
T_{t-1}	Smoothed additive trend at the end of period t
x_t	Currently observed demand value

Introduction

Aircraft maintenance is subject to stringent regulations, with safety being the absolute top priority. Both aircraft carriers and Maintenance, Repair, and Overhaul (MRO) organizations aim to optimize maintenance procedures to minimize aircraft downtime [2, 20]. Over the past decade, numerous machine learning algorithms have been developed and integrated to improve maintenance management. Aircraft are now equipped with specialized sensors, leading to increased data collection regarding component failures [21]. However, while MROs and aircraft carriers have made progress in predicting component failures, they have difficulty obtaining a complete list of materials for dedicated maintenance events due to the complexities of technical documentation. Consequently, the spare parts required to complete maintenance events are often unavailable during the execution phase, resulting in unexpected delays and increased costs.

Purchasing these spare parts is not as straightforward as expected. Aircraft consist of thousands of unique components that are typically purchased from aftermarket companies, which are distributors of aircraft spare parts. The availability of aircraft spare parts at aftermarket distributors may differ; some spare parts are on the shelf and can be delivered directly, while other spare parts may be very case specific and need to be produced or ordered from suppliers first, resulting in a delivery time of months, which substantially affects the duration of the maintenance event. This underscores the importance of having a comprehensive material list in advance to streamline planning, improve efficiency, and minimize aircraft downtime. Additionally, predicting future demand for spare parts poses challenges for aftermarket distributors due to the distinct characteristics and variable demand patterns associated with each spare part. Traditional demand prediction models rely on historical sales records to forecast monthly demand values.

This thesis is done in collaboration with an aircraft aftermarket distributor and addresses the aforementioned challenges by developing a robust prediction model. This model forecasts the occurrence of customer-specific purchases of Maintenance Planning Document (MPD) related spare parts by incorporating technical documentation and historical sales records. All necessary information, including the required sales records and identified MPD related spare parts, was provided by the aftermarket distributor. To the best of the author's knowledge, this paper marks the first attempt to integrate these components into a single architecture tailored to the needs of the aftermarket distributor. Such an algorithm has the potential to revolutionize the aftermarket industry, shifting it from a reactive environment to a proactive one. For example, it could be implemented as a spare part recommendation system, introduces the possibility of dynamic pricing, and can be utilized to optimize stock levels and improve current supply chain processes.

This thesis report is organized as follows. In Part I, the scientific research paper is presented. This is the final and main contribution of this thesis and discusses the methodology, results, limitations, and potential applications of the robust prediction model. The second part of the thesis report, Part II, covers the literature review that was completed as the initial step in this thesis. It summarizes previous research on aircraft spare parts and the use of machine learning models for demand prediction. Finally, in Part III, a comprehensive overview of the results obtained from two case studies that assess the proposed classification model described in the scientific paper is provided.

I

Scientific Paper

Predicting Future Aircraft Spare Part Purchases by Using Previous Sales Records and Technical Maintenance Documentation

M.L. Wittenberg,*

Delft University of Technology, Delft, The Netherlands

Abstract

Efficient maintenance scheduling is a critical objective for aircraft carriers and Maintenance Repair and Overhaul Organizations (MROs) to minimize aircraft downtime. While predictive maintenance models have improved, accurately identifying materials, especially spare parts, for specific maintenance events remains challenging. This paper combines the challenges of identifying spare parts by MROs and aftermarket distributor demand models by developing a robust prediction model (SPSO-CM) to forecast subsequent customer-specific purchases of maintenance planning document (MPD) related spare parts, considering technical documentation and previous sales records. The proposed architecture employs a gradient-boosting algorithm with numerous data mining improvement techniques to predict the likelihood of a subsequent spare part purchase from a customer. A k-means clustering algorithm is used to group spare parts with similar characteristics, as certain specific spare part properties significantly influence demand prediction models. A unique feature selector and nested group K-fold TimeSeriesplit cross-validation method were developed and incorporated into the Bayesian search space to optimize hyperparameters and improve performance. Two test cases were simulated, and the results demonstrated that SPSO-CM is more effective in forecasting proprietary parts and frequently purchased spare parts than those with extremely lumpy demand patterns. Two potential applications for an aftermarket distributor are discussed, one from a customer-level perspective and another at a larger supply-chain level, highlighting its promising capabilities.

keywords: Aircraft aftermarket distributor, aircraft maintenance, classification model, demand prediction, gradient-boosting, machine-learning

1 Introduction

The aviation industry has grown rapidly in the last decade. Millions of passengers travel every day with thousands of competing airlines that use different aircraft types. Current global market forecasts predict that passenger and freight traffic will even further increase annually by 3.6% and 3.2%, respectively, towards 2040 [1]. To accommodate this growth, airliners must optimize their current operational fleet usage by planning maintenance activities as efficiently as possible to limit downtime of their aircraft while keeping passenger safety as a key priority.

Maintenance, Repair, and Overhaul (MRO) organizations and aircraft carriers have traditionally relied on maintenance regulations, component data, and past maintenance schedules to predict and schedule maintenance activities [2]. However, accurately identifying the materials required for a specific maintenance event, including spare parts for related events, remains a significant challenge. Incomplete material bills can lead to unexpected delays during maintenance, resulting in high downtime costs. Purchasing these spare parts is not as straightforward as expected. Aircraft consist of thousands of unique components that are typically purchased from aftermarket companies, which are distributors of aircraft spare parts. The availability of aircraft spare parts at aftermarket distributors may differ; some spare parts are on the shelf and can be delivered directly, while other spare parts may be very case-specific and need to be produced first, resulting in a delivery time of months which substantially affects the duration of the maintenance event. This challenge became even more pressing after the COVID-19 pandemic, when many grounded aircraft had to be reintegrated into service and delayed maintenance tasks had to be quickly addressed, leading to a high demand for aircraft spare parts [3]. A recent news article by van der Heide [4] on current spare part delivery issues caused by problems in the Boeing and Airbus supply chain illustrates these challenges.

Forecasting demand for aircraft spare parts is challenging for aftermarket distributors due to the extreme number of different spare parts, each of which exhibits unique characteristics and varying demand patterns. Aircraft components are frequently classified as Rotables, Repairables, Expendables, or Consumables based on their economic value, functionality, essentiality, and expected lifespan. Rotables, including landing gears and major engine components, are designed for unlimited repairs and will not be replaced under normal operating conditions. Repairables, such as engine blades and tires, have a limited repair lifespan before replacement is necessary. Expendables, such as cotter pins

*Msc Student, Air Transport & Operation, Faculty of Aerospace Engineering, Delft University of Technology

and rivets, cannot be repaired and are regularly replaced with new parts. Consumable items, like oil and lubrication, are depleted during aircraft operations and require periodic refreshment [5, 6].

Traditionally, aftermarket demand prediction models are based on previous sales records to predict monthly demand values to opt in spare part inventories, which leads to lower holding costs and increased customer satisfaction. This research combines the challenges of identifying spare parts needed by MROs and aftermarket demand models by developing a robust prediction model to forecast subsequent customer-specific purchases of maintenance planning document (MPD) related expendable spare parts considering technical documentation and previous sales records. The proposed architecture employs a gradient-boosting algorithm with numerous data mining improvement techniques to predict the likelihood of a subsequent spare part purchase from a customer. To the best of the author's knowledge, this paper is the first to combine technical aircraft documentation with an aftermarket distributor's sales records to predict the occurrence of future purchases. This model empowers aftermarket distributors to shift their business strategy from a reactive approach to a more proactive one, enabling them to recommend spare parts to customers and take preemptive action before specific purchases occur.

This paper is structured as follows: A review of the relevant literature on aircraft spare part demand prediction models is presented in Section 2. Subsequently, Section 3 provides a comprehensive explanation of the problem, the objective, the setup, and the underlying assumptions. A detailed description of the proposed model's architecture is presented in Section 4. Various framework configurations are evaluated to identify the best-performing algorithm by considering two case studies, and the results are discussed in Section 5. The initial steps toward developing a use case application utilizing the spare part subsequent purchase order occurrence prediction model are discussed in Section 6. Finally, the conclusion of this article, the limitations of the study, and the recommendation for future research are discussed in Section 7.

2 Literature Review

This section provides an overview of related work, including forecasting models designed to address the specific characteristics of aircraft spare parts in Subsection 2.1, as well as data improvement techniques discussed in Subsection 2.2 to improve forecasting models.

2.1 Forecasting Models

Aircraft spare parts exhibit a unique and uncertain demand pattern, necessitating the classification of demand patterns to select the most suitable prediction model for accurate results. The commonly used Syntetos-Boylan-Croston modified Williams scheme categorizes spare parts into smooth, erratic, intermittent, or lumpy classes based on their average demand interval (ADI) and squared correlation of variation (CV^2) [7]. Aircraft spare parts typically exhibit an intermittent or lumpy pattern, characterized by a high average demand interval. This uncertainty in demand quantity and interval poses significant challenges for demand forecasting and limits the choice of prediction models [2].

Various forecasting methods have been proposed in the existing literature to cope with uncertain demand quantity and irregular intervals. The Croston method [8], widely used in the aviation industry to address lumpy demand behavior, employs two separate single exponential smoothing (SES) models: one to predict demand sizes and the other to predict demand intervals. Several studies compared the performance and precision of the Croston method with a single SES model to predict aircraft spare parts demand.

Regattieri et al. [9] compared parametric forecasting techniques using historical demand data from A320 spare parts used by Alitalia, which exhibited a lumpy demand pattern. The study demonstrated the superiority of the Croston method over simpler SES approaches, indicated by the Mean Absolute Deviation Average (MAD/A). Similarly, in another study by Eaves and Kingsman [10] on predicting aircraft spare part demand for the UK Royal Air Force, the Croston method outperformed simpler single-parametric approaches.

The Syntetos Boylan Approximation (SBA) [11] is a modified version of the Croston method. The original model was found to be biased, as it assumed the independence between the predicted results for the demand interval and the size of the demand. The researchers introduced a correction term in the SBA version to address this bias. Testing the modified SBA model on an automotive demand dataset containing 3000 intermittent pattern spare parts, it demonstrated superior performance over the original Croston method, measured by the scaled mean error and Geometric Root Mean Square Error (GRMSE). Despite its bias, the Croston method remains widely used in the aviation industry due to its simplicity, relatively good results, easy implementation, and wide availability in leading software packages [11].

Parametric approaches, despite their usefulness, have limitations in recognizing nonlinear patterns, leading to misinterpretation of relationships between dependent and independent variables. To overcome this, nonparametric models such as artificial neural networks (ANN) are proposed in research papers to forecast spare part demand [12]. A commonly used ANN form is the Multi-Layered-Perceptron (MLP) network with Back Propagation (BP) algorithm. Gutierrez et al. [13] developed a 3-layer MLP model to forecast daily demand for 24 electronic spare parts that exhibit a lumpy demand pattern. Their model considered the demand value at the end of the immediately preceding time period and the time difference between the previous two non-zero demand transactions. Comparing the MLP performances with three parametric models (SES, Croston, and SBA), the proposed MLP outperformed the parametric approaches in terms of MAPE when the ADI showed no significant difference between training and testing data. However, in cases where a significant difference was present, the parametric models achieved a slightly lower MAPE than the proposed MLP model.

Extending the work of Gutierrez et al. [13], Babai et al. [14] further improved the earlier proposed MLP by incorporating additional input parameters from previous time periods. Analyzing their model on a real-case dataset containing 5000 aircraft spare parts from an airliner, they concluded that the extended MLP model reached higher forecast accuracy compared to the same three parametric approaches and even surpassed the performance of the original MLP model Gutierrez et al. [13]. Furthermore, Amirkolaii et al. [15] constructed an MLP to predict the demand for 30 different spare parts, exhibiting a lumpy demand pattern, for a French Aircraft Original Equipment Manufacturer (OEM) used in business aircraft. Various input features were utilized, including the previous demand interval, ADI, CV^2, the number of periods separating the previous two non-zero demand intervals, and the price of the spare part. Their results clearly demonstrated that the proposed MLP outperforms Croston and SBA.

An alternative ANN model that incorporates prior information to forecast spare part demand is the Recurrent Neural Network (RNN). Unlike the MLP network, the RNN has an extra layer that sends the output of the model back into the input layer [16]. Unfortunately, the RNN can be affected by the vanishing exploding gradient problem, leading to unrealistic results [17, 16]. To address this problem, Hochreiter and Schmidhuber [18] introduced the Long Short-Term Memory Network (LSTM), which equips each neuron with a memory cell featuring input, output, and forget gates to effectively manage the information flow. Renowned for its versatility and strong learning capabilities, high-tech companies such as Google, Facebook, and Amazon have widely adopted the LSTM network for translation and speech recognition products [19]. Recently, these models have gained significant interest in time-series predictions.

Chandriah and Naraganahalli [20] employed a 6-layer RNN-LSTM model to forecast the demand for passenger cars in Norway and found that it outperformed standard parametric approaches (Croston, SES, SBA) in terms of Mean Squared Errors when predicting lumpy/intermittent demand patterns.

Possible downsides of ANNs include their lack of interpretability, often referred to as black boxes, as well as their substantial need for training data, which may not always be readily available. This issue becomes particularly pertinent when dealing with complex systems such as aircraft that consist of thousands of long-lifecycles components. A promising solution to these forecasting challenges comes in the form of Gradient Boosting models. These supervised machine-learning models ensemble multiple weak decision trees to construct a stronger robust prediction model. Gradient Boosting algorithms are renowned for effectively managing mixed data types, strong performances despite limited data availability, and run-time efficiency.

In a study by Chang and Meneguzzi [21], the predictive performance of an extreme gradient boosting (XGBoost) model was compared to an MLP network and other machine learning regressors using a computer retailer sales dataset. The considered input features included a limited subset of contextual data and some geographical information from the customer. The results indicated that the XGBoost model outperformed or matched the other selected approaches in terms of performance metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Square Error (MASE), and MAPE. Furthermore, the researchers validated the efficacy of the model using a bike-sharing demand dataset and consistently observed enhanced predictive capabilities across both datasets.

In summary, based on the existing literature, various forecasting techniques can be utilized for demand predictions. However, it is evident that ANNs and gradient boosting machine-learning algorithms outperform traditional parametric approaches when dealing with lumpy or intermittent demand patterns. The literature review also concluded that feature engineering is crucial to improve the performance of the prediction model; additional literature on this topic is provided in Subsection 2.2. Based on the described pros and cons, this study decided to employ a gradient-boosting forecasting algorithm due to its limited data effectiveness and interpretability.

2.2 Data Mining Improvement Strategies

Numerous studies state that the effective utilization of big data mining techniques significantly enhances the accuracy of prediction models [22]. This research investigates the application of two big data mining techniques, association rule mining (ARM) and unsupervised clustering, in combination with demand forecasting.

Traditional ARM techniques, often referred to as the market-basket approach, discover correlations between non-consecutive items within large datasets [23]. This frequent item set discovery process is commonly achieved through the A-Priori algorithm Agrawal and Srikant [24] and the FPgrowth algorithm [25]. Although both techniques aim to yield the same frequent itemsets, they differ in how they construct search spaces, leading to variations in runtime efficiency and memory usage. A comparative analysis conducted by Singh et al. [26] affirmed the equivalence of the association rules derived from both algorithms. However, when dealing with larger databases, the FPgrowth algorithm converges more quickly and consumes less memory than the A-priori algorithm. Conversely, for smaller databases, the A-priori remains preferable due to its scalability.

Chen and Wu [27] explored the impact of ARM on the efficiency of order/batch picking of products in warehouses. Their research employed the A-priori algorithm to uncover correlations among different orders, resulting in a significant increase in order-picking efficiency in warehouse operations. Another study conducted by Moharana and Sarmah [28], examined the implications on stock levels and inventory control by considering frequently co-occurring sold spare parts as cohesive groups rather than individual items. The findings of their study showed that interdependencies between spare parts lead to reduced stock levels and improved inventory control. Furthermore, the researchers highlighted how shortages in critical components can significantly influence the shortage and holding costs related to the interconnected minor parts.

Another interesting big data mining technique that may improve the performance of demand forecasting models is unsupervised clustering. Clustering techniques aim to categorize data points, such as customers with similar characteristics and purchase behavior. These segments are primarily used to develop marketing strategies within the e-commerce sector. However, recent studies also examined the effects of data segments on forecasting models.

In the study conducted by Kalchschmidt et al. [29], distinct forecasting models were trained for individual customer segments. Subsequently, the researchers amalgamated the results of all distinct groups to project the total demand. In contrast, the methodology adopted by Caniato et al. [30] focused solely on training the forecasting model to predict the demand of the customer located closest to the centroid of the cluster. Following this step, the obtained forecast results were extrapolated to the entire cluster, resulting in an overall demand forecast. Customer segmentation was achieved by considering a diverse set of historical demand variables that describe purchase behavior, followed by the application of K-means clustering. Despite the different approaches, both studies demonstrated that by initially clustering demand patterns, forecasting models can be improved.

3 Problem Description

This section formulates the problem and the research objective. First, a short introduction to aircraft maintenance and aftermarket distributors is given in Subsection 3.1. The research objective of this study is stated in Subsection 3.2. Followed by a detailed description of the problem setup in Subsection 3.3, including the data sources to construct the input of the model and the projected output. Finally, the assumptions to delineate and refine the scope of this study are presented in Subsection 3.4.

3.1 Aircraft Maintenance Aftermarket Order Predictions

Many aircraft maintenance prediction models are designed to predict aircraft component failure. These predictions are based on historical maintenance activities, degradation data, aircraft utilization rates, and regulatory documentation for aircraft maintenance. Maintenance Repair and Overhaul organizations (MROs) utilize these predictions to enhance fleet schedules and minimize (unexpected) aircraft downtime [2]. However, these models often overlook the execution complexity of the projected maintenance events.

Each aircraft maintenance event is documented within a Maintenance Planning Document (MPD), containing the event's maximum utilization threshold, denoted in Flight Hours (FH), Calendar Days (CD) or Flight Cycles (FC), number of required mechanics, average completion time per mechanic, and the corresponding Aircraft Maintenance Manual (AMM) reference number [31]. The AMM, in turn, provides a detailed step-by-step procedure for the execution phase; a comprehensive overview of conditional and on-conditional required spare parts; Essential tools/equipment; and any interconnected AMM tasks that must be accomplished before or after the originally scheduled maintenance event. In essence, not only the ability to predict the need for a maintenance event, but also gaining a comprehensive understanding of its execution phase, including the availability of necessary spare parts, is essential in optimizing fleet schedules and minimizing any additional downtime due to maintenance activities.

The detailed step-by-step procedure and references to related AMM tasks can make it difficult for MROs and aircraft carriers to prepare a complete material list in advance for a specific maintenance event. Experienced mechanics and maintenance schedulers are better equipped to understand the process and ensure that all necessary parts are available in advance. However, less experienced personnel may struggle with this task. Additionally, some maintenance events are infrequent, occurring only once every few years, making it even harder to create a comprehensive materials list due to the lack of prior experience and knowledge. Having a complete bill of materials in advance is crucial for the purchasing process at an aftermarket distributor. Not all spare parts are readily available; some are rare or specific to certain aircraft configurations. In such cases, the aftermarket distributor must initiate a back-order process with the original spare part supplier. In the worst-case scenarios, this can introduce delays of several months. Consequently, the necessary spare parts cannot be delivered immediately to the MRO, leading to aircraft unavailability.

Forecasting demand is challenging for aftermarket distributors due to the varying properties of spare parts and the lumpy demand pattern. Traditionally, these models relied on past monthly sales volumes, lacking specificity in customer considerations and technical spare part relationships detailed in the AMM. Aftermarket distributors that are integrated with an Aircraft OEM, on the other hand, can combine technical knowledge from maintenance documentation with previous sales records of spare parts to predict upcoming purchase orders. These predictions can be utilized to construct a complete bill of material for an aftermarket customer. Additionally, these predictions can optimize aftermarket inventories. Furthermore, it has the potential to transform the entire aftermarket distributor business model. Currently, the business model is reactive, responding to customer needs as they arise. However, if successful, the prediction of subsequent spare part purchase orders introduces the potential for a proactive approach, allowing the aftermarket to anticipate the future needs of the customer.

3.2 Research Objective

The objective of this research is to develop a robust forecasting model to predict the occurrence of subsequent purchase orders of spare parts related to the same MPD event to enhance inventory efficiency at aftermarket distributors. To achieve this objective, a machine-learning classification model is developed.

The classification model predicts spare part order occurrences using historical transaction records, technical aircraft maintenance documentation, and available spare part characteristics translated into various input features. These features are transformed via different preprocessing steps, such as scaling methods and encoders, into the required model's input format. A feature selector algorithm is constructed to select a subset of features, based on the correlation with the projected output, during the model's training and tuning phase to optimize the performances. The predicted outcome is compared with actual spare part occurrences, and the results are evaluated via various classification error metrics.

In addition to developing the classification model, this paper explores its potential applications for the aftermarket distributor. These applications range from generating a comprehensive customer-level bill of materials for specific maintenance events to optimizing inventory levels at a broader supply chain level.

3.3 Research Setup

This research is conducted in close collaboration with an aftermarket distributor that is integrated into an aircraft OEM. This collaboration offers access to multiple data sources necessary for the development of the classification model. An overview of these data sources is presented below:

- **Sales Records:** The sales records cover all available transactions of the aftermarket distributor. Each order line comprises a date stamp, customer name, unique part ID, ordered quantity, order reference number, paid price, priority code, and order placement warehouse. The order priority codes classify the transactions into four distinct priorities: Aircraft On Ground (AOG), Work Stoppage (WSP), Routine (RTN), and Urgent Stock Replenishment (USR). AOG and USR represent unexpected urgent orders, while WSP and RTN denote planned orders by customers.
- **Technical Documentation Forecast (TDF) Information:** The TDF information encompasses the outcomes of a currently in-development tool within the company. This algorithm provides a comprehensive overview of all relevant spare parts associated with a selected MPD event. By using the MPD event code, the tool searches for the corresponding AMM reference number. Subsequently, it generates a complete list that includes tools/equipment, expendables, and consumables mentioned in the context of the AMM. In particular, it outlines the conditional necessity for these components along with the spare part category code and the essentiality code. Moreover, if additional AMM references are identified in the context, whether preceding or succeeding the original (first-level) AMM task description, the tool incorporates these supplementary tools and spare parts. This process extends to a maximum AMM depth of three levels.
- **Spare Part Catalog:** This catalog describes the interconnection among spare parts. It provides a comprehensive overview of all possible interchangeable spare parts related to the selected component, along with the part's evolution over time.

3.4 Assumptions

Numerous simplifications and assumptions were essential to enhance this research to simplify data analysis and develop the model for analyzing spare part orders.

Upon analyzing sales records, inconsistencies were identified in the use of purchase order reference numbers by customers over time. It was found that nearly all purchase orders consisted of a single purchase order line, suggesting that different spare parts are rarely sold together. The company experts supported this data observation. Consequently, it is assumed that each purchase order line represents a distinct purchase order, treating different spare parts bought by the same customer on the same day, with similar or dissimilar order reference numbers, as separate orders. Additionally, to simplify the problem, separate purchase orders of the same spare part on the same date by the same customer with identical order line properties are consolidated into one single large purchase order line by aggregating quantities and turnover. This consolidation assumes that a customer may have placed a purchase order with lower than intended quantities. Similarly, when forecasting future purchase order occurrences, the model treats scenarios in which a particular spare part is ordered multiple times within a defined time frame as a singular occurrence. This approach is rooted in the fact that the purchase order occurrence classification model exclusively predicts the instances of occurrences, without delving into quantity predictions associated with these projected occurrences. This simplification aligns with the model's exclusive emphasis on forecasting the probability of subsequent spare part purchase order occurrences, regardless of quantities or number of separate upcoming purchase orders.

Given that each order line only has a date stamp, it is impossible to discover the chronological order of purchase orders on a specific date. Therefore, it is assumed that the sorted sales record database represents the chronological order of transactions. Furthermore, in terms of purchase order priorities, it is assumed that every customer truthfully selects the priority. This assumption is necessary since company experts noted that customers occasionally opt for a higher priority for faster delivery times, despite the absence of an actual AOG scenario.

Additionally, assumptions are made regarding the TDF documentation and MPD events. Each aircraft, even of the same type, possesses a unique MPD and AMM due to factors like customer preferences, production year variations, and advancements in construction methods. Consequently, predicting subsequent spare part purchase order occurrences based on similar MPD events becomes challenging, as similar MPD events may require different spare parts. Therefore, to make this research feasible, it is assumed that the spare parts identified for an MPD event by the TDF remain consistent across all aircraft. Furthermore, it should be noted that the influence of spare parts shared among multiple MPD events is not considered in the classification model for simplicity.

Finally, in terms of interchangeability between spare parts, it is assumed that all interchangeable spare parts, whether one-way, two-way, or predecessors of the original parts listed on the TDF, can be used for the same MPD event. The reason for this assumption is to include all possible spare parts purchase order relationships in the analysis to improve the predictive performance of the classification model.

4 Methodology

To fulfill the objective presented in Section 3, a Python-based model named the Spare Part Subsequent Purchase Order Occurrence Classification Model (SPSO-CM) has been developed. The SPSO-CM is designed to classify subsequent purchase order occurrences for spare parts associated with the same MPD event. It employs a single-label stacked classification approach, where each label corresponds to all subsequent purchase orders for a specific spare part and is independently trained. Once trained, the predicted purchase order occurrences for all spare parts are stacked together and evaluated as one unit for an entire MPD event.

The SPSO-CM architecture is presented in Figure 1. This independent architecture consists of seven main components:

- *Input Parameters*: A dictionary containing the required input parameters for the SPSO-CM.
- *Database Collector*: The necessary database is constructed, including transaction records, TDF information, and spare part catalog, based on the predefined input parameters.
- *PN Clustering algorithm*: All listed PNs in the Available Information Library are analyzed and grouped according to their similarities.
- *Feature Creator*: MPD-specific input features and projected target labels are generated from the Available Information Library.
- *Preprocessor*: Processing the created features for the prediction model.
- *Prediction Model*: Optimization and fine-tuning of the prediction model.
- *Evaluation*: Assessing the performance accuracy of the complete SPSO-CM architecture.

The SPSO-CM operates based on specific input parameters, including MPD event codes, time lag (tl), time-window (tw), and the selected configuration. These parameters guide the functioning of the six remaining components of SPSO-CM, each of which is detailed in this section. Starting with the database collector in Subsection 4.1, followed by the PN clustering algorithm in Subsection 4.2. The MPD-specific features are constructed in Subsection 4.3. The descriptions of the preprocessor and the employed gradient-boosting algorithm can be found in Subsection 4.4 and Subsection 4.5, respectively. Finally, the section concludes by examining the performance of SPSO-CM in Subsection 4.6.

4.1 Database Collector

The database collector imports commas-separated value (CSV) files containing the company's sales records and spare part catalog. Given the input MPD event codes, the company's TDF tool extracts conditional and unconditional expendable spare parts from the MPD and AMM documentation. These parts are extracted from the TDF's output and stored in a new spare part database, known as the PN nomenclature. Each spare part in this nomenclature receives a new unique part ID. The PN nomenclature is expanded to include all interchangeable parts from the catalog, each inheriting the original part's ID. Purchase orders with nomenclature-listed spares are isolated from sales records, after which the original part IDs are replaced by the corresponding nomenclature spare's ID. The remaining purchase orders are then sorted by purchase date and customer name and stored, together with the spare part nomenclature, in the available information library.

4.2 PN Cluster algorithm

Based on the literature review in Section 2 it became evident that each spare part exhibits different properties and therefore it is difficult to utilize one forecast model that works for all PNs. To address this challenge, numerous spare parts properties are derived and used to cluster PNs with similar characteristics. These variables, along with the derived PN cluster ID (C_{PN}), can then be used as additional input features for the prediction model. Moreover, the derived clusters help

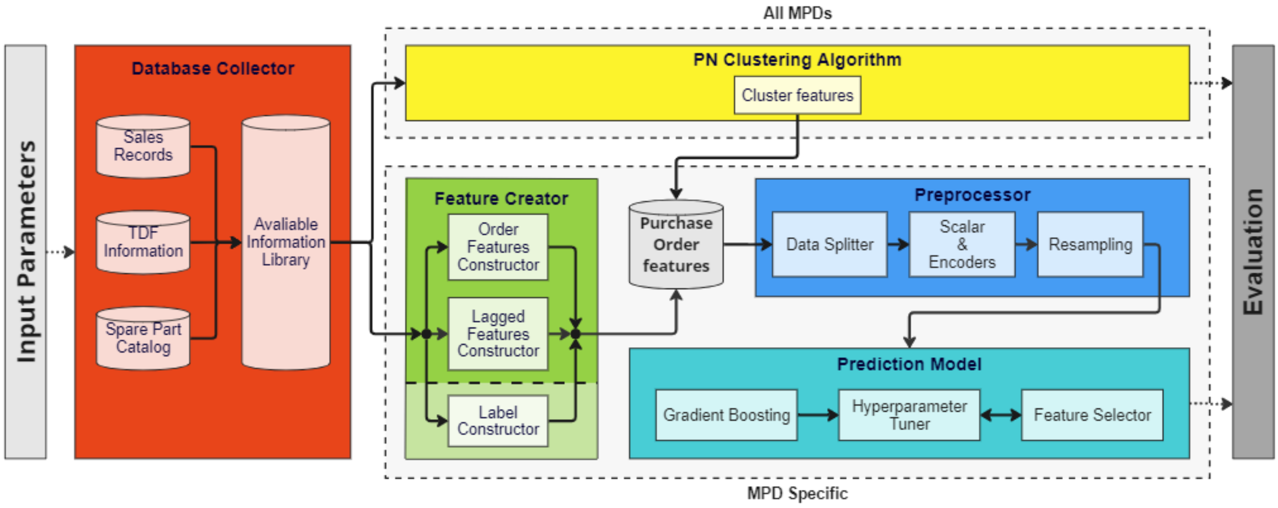


Figure 1: Visual representation of the SPSO-CM's architecture.

in identifying the suitability of the SPSO-CM for specific PNs. An overview of the derived PN properties from sales records and TDF documentation is listed below:

- *Material Group* (Mat_{PN}): A numerically encoded variable that characterizes the type of the PN. A value of 30 represents vendor parts, 70 corresponds to standard parts, and 10 indicates proprietary parts.
- *Average Quantity* ($\overline{Qnt}_{PN}$): The average amount of PN sold per purchase order.
- *Number of Interchangeable PN* ($N_{PN-inter}$): Number of interchangeable items related to the respective PN.
- *PN support* (N_{PN}): Total number of purchase orders associated with the specified PN.
- *PN's unique customers* ($N_{PN,customers}$): The number of unique customers who purchased the PN.
- *Average Demand Interval* (ADI_{PN}): Mean time between consecutive purchase orders, calculated in months.
- *Squared Correlation of Variation* (CV^2_{PN}): Squared correlation of the monthly variation in the quantity of the PN.

A k-means clustering algorithm is employed on the listed PN properties to identify which PNs have similar characteristics. This algorithm, imported from the Sklearn Python library, initially divides data points into K clusters and then iteratively reassigns them to minimize the squared distances between data points and cluster centroids until an optimum is achieved [32]. As the K-means algorithm requires the number of optimal clusters as an input parameter, it should first be determined by computing the silhouette score and constructing the elbow method. The silhouette score measures inter-cluster separation, with higher scores indicating better clustering. While the elbow method plots the sum of squared distances to cluster centroids (WCSS), identifying a turning point where further splitting becomes less effective, revealing the optimal cluster count. By iteratively applying the k-means algorithm with varying K values, an optimal number of clusters is found and used to cluster the MPD's PNs.

Eventually, the listed PN input features and C_{PN} are substituted in the PN nomenclature, where they can be extracted as additional SPSO-CM input features.

4.3 Feature Creator

The remaining MPD-specific purchase orders in the available information library are utilized to form input features for the classification model and generate the model's training output. The feature creator consists of three parallel feature construction blocks: *Order feature Constructor* generates input features from the respective purchase order line; *Lagged feature Constructor* constructs input features from information prior to the current purchase order line; And the *Label Constructor* generates the target label that indicates future spare part purchase order occurrences, necessary for training and evaluating the classification model. All these features are created for every MPD-specific purchase order line within the available information database. A more detailed explanation of each feature creator block is given below.

Order Features Constructor

Each purchase order line (X_t) contains predictive information about future spare parts purchase orders. The order input features extracted from the X_t are Spare Part ID (PN), Customer Name ($customer$), Order priority code ($priority$), order month ($month$), order year ($year$), ordered quantity of the corresponding PN (Qnt_{X_t}), and the warehouse where the purchase order is placed (Wh_{X_t}). It is important to note that $month$, $year$, and Qnt_{X_t} are presented as numerical properties, while PN, $customer$, and $priority$ are categorical properties.

Lagged Features Constructor

Lagged features capture valuable information from preceding purchase orders ($X_{PN,t-1}$) leading up to the current purchase order (X_t). As highlighted in the literature review, these features can significantly enhance the performance of prediction models. The classification model incorporates four key lagged input features:

- *previous time* (ΔT_{PN}): Signifies the time interval in days between X_t and $X_{PN,t-1}$.
- *previous quantity* ($Qnt_{\Delta T_{PN}}$): Denotes the associated quantities of each $X_{PN,t-1}$.
- *number previous purchase orders* ($N_{X_{PN,t}}$): represents the frequency of the previous purchase orders of the customer of the specific PN.
- *adjacent set*: represents the previously purchased PNs within the time lag precedent to the X_t.
- *adjacent set support* ($F_{adjacent}$): Equals the frequency of the adjacent set in the database.

The extraction principles and characteristics of ΔT_{PN}, $Qnt_{\Delta T_{PN}}$, and $N_{X_{PN,t}}$ are the same. These three input features are PN-specific, thereby separately computed for each PN within the MPD event. Moreover, their representation varies, being either integers or Not a Number (NaN) values, depending on whether the customer had previously purchased the PN.

The remaining two lagged features, *adjacent set* and $F_{adjacent}$, are inspired by ARM techniques. By using the predefined time lag input parameter, an adjacent set of previous purchase orders is created. This variable is encoded in a binary chromosome format, where each segment of the chromosome corresponds to a specified PN. A value of 0 signifies the absence of the PN, while 1 indicates its presence within the predefined time lag. To illustrate, the presence of PN A.1 and PN A.4 in MPD A which consists of 5 parts is denoted as $\{10010\}$.

A supplementary computational step is required to calculate the associated support value, $F_{adjacent}$. A rolling window, equivalent to the predefined time lag, is applied to each X_t to generate a feasible set of purchase order baskets per customer. Subsequently, the FP growth algorithm is utilized to compute the support values of all potential frequent itemsets. This process is done per available order year, encompassing all baskets up to that year. As a result, the first available order year in the database does not contain any frequent itemsets. However, for subsequent order years, all records from the preceding year(s) are included. This cumulative process establishes a comprehensive database of support values for all frequent itemsets leading up to each available order year. Ultimately, $F_{adjacent}$ can be extracted from this database by looking up the support value of the early defined *adjacent set*.

Label Constructor

The target labels necessary to train and evaluate the SPSO-CM are derived along with the input features. For each X_t, the label constructor extracts all upcoming purchase orders from the same customer within the predefined tw. As the SPSO-CM is a classification model, the target labels ($y_{PN,t+tw}$) are binary encoded: 0 indicating no future purchase orders of a specific PN in the subset, while 1 signifies that the customer has purchased the specified PN in the upcoming tw.

4.4 Preprocessor

The final step in the SPSO-CM before employing the classification model is the Preprocessor block. This part of the architecture ensures that the input data set is ready for the training and evaluation phase. First, the input features, together with the binary encoded target labels, are split into a training and testing set by the *Time Serie Split* processor, after which the encoder preprocesses all features into suitable formats for the classification model. The last step in the preprocessor undersamples the train data to deal with the imbalances between occurring and non-occurring subsequent purchase orders.

Data split

The first stage of the SPSO-CM preprocessor involves dividing the computed X_t features, which have been sorted in a timely order, into a separate training and test subset. This split is essential for the subsequent training and evaluation phases of the SPSO-CM framework and is shown in Figure 2.

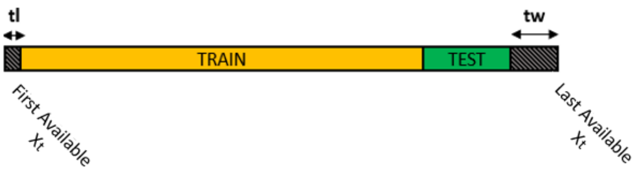


Figure 2: The initial split of the available X_t into a training and test set.

Before partitioning X_t into a train and test set, an additional data exclusion step is applied to maintain the integrity of the input features aligned with the predefined tl and tw, highlighted in gray in Figure 2. As described in Subsection 4.3, lagged input features are derived using purchase orders prior to the analyzed X_t. Consequently, initial X_t that lack the necessary information to compute lagged features are disregarded. Similarly, X_t at the tail end of the dataset are eliminated, as generating target labels beyond this period is not feasible. To guarantee sufficient training and test data for a robust performance assessment, the remaining portion of the dataset is split using Sklearn's timeseriesplit algorithm, employing an 85% training and 15% testing split ratio.

Scalars & Encoders

Most machine-learning classification models require feature encoding before training to accommodate unsupported data types and potentially improve model performance. The selected gradient-boosting classification models, which will be discussed in Subsection 4.5, specify that decision trees do not require numerical scalars. However, assessing the scalar's impact on the model performance is recommended. Categorical feature types, on the other hand, are not supported by selected classifiers and require numerical conversion.

As described in Subsection 4.3, the input features for each X_t encompass both numerical and categorical types. To address this, the SPSO-CM framework integrates three distinct feature encoders:

- **MinMax Scalar:** A data normalization technique that scales numerical values to a 0-to-1 range, ensuring uniformity among various input features and leading to enhanced model performance.
- **One Hot Encoder:** A categorical transformation technique that converts categorical features into binary values. For each category, an additional/separate input feature is created to indicate its presence within X_t. While this encoder is effective when dealing with a few categories, it loses effectiveness when numerous categories exist, causing higher input feature dimensionality and data sparsity. Consequently, more training time is required, and the risk of overfitting increases.
- **Target Encoder:** This categorical encoding technique substitutes the original category with its calculated impact on the corresponding target labels. The impact is determined by computing the posterior probability of the target label's presence given the respective category. Unlike the One Hot Encoder, this approach does not alter the dimensionality or sparsity of the input feature since no extra features are introduced.

The encoders employed for each input feature are specified within the SPSO-CM's input parameters. Various combinations between input features and encoders are assessed to achieve optimal performance. Nevertheless, categorical features with more than 10 categories are only target encoded to control dimensionality and data sparsity.

Resampling

The Near-Miss version 3 undersampling technique is used to address the imbalance within the training data. This method alters the composition of the training set by selectively eliminating certain data points from the majority class, typically $X_t|y = 0$, to establish a balanced ratio between $X_t|y = 0$ and $X_t|y = 1$. The version-3 variant only selects majority class examples that are closest to each minority class data point, resulting in a balanced dataset with $X_t|y = 0$ situated close to the decision boundaries.

4.5 Prediction Model

Following the preprocessing of input features and the creation of a training and test dataset, the next phase of the SPSO-CM architecture initiates the training phase of the classification model. Within this context, two gradient-boosting classification algorithms are examined and incorporated into the SPSO-CM model as part of this study. The differences between these models, together with a better explanation of gradient-boosting models, are discussed in this section. Furthermore, fine-tuning the hyperparameters of the chosen classifier through a Bayesian search optimization method is described. To further enhance performance, a self-developed feature selector is introduced and integrated into the Bayesian search space.

Gradient Boosting

Gradient boosting is a machine-learning model that combines multiple weak decision trees to make better predictions. The ensembling process of poor learners, typically decision trees, is called boosting. The process starts with fitting an initial decision tree to the data. Subsequently, a sequence of trees is generated, each of which focuses on the data points where the preceding model exhibited poor performance. This iterative process strengthens the ensemble model, leading to more robust predictions. The gradient term in gradient boosting refers to the way that additional trees are created. These successive trees are designed to capture the maximum variance between the target values (y) and the predicted values ($\hat{y}$), which can be mathematically expressed as the partial derivative of the loss function: $(-\frac{\partial L}{\partial \hat{y}} = y - \hat{y})$. By utilizing this computed gradient as the target for new trees, the model optimizes its performance by explaining the maximum amount of variation within the overall gradient-boosting model [33]. The interested reader may refer to the work of Bentéjac et al. [34], for more information on the construction of gradient-boosting models.

Over the years, many gradient-boosting variants have been developed for regression and classification tasks. Two of these algorithms, XGBoost and LightGBM, are considered for the SPSO-CM's framework. A detailed explanation of their properties and underlying differences is listed below [33]:

- **XGBoost:** The XGBoost algorithm, developed by Chen and Guestrin [35], is one of the most well-known gradient-boosting methods and often serves as the primary choice by many practitioners to quickly solve prediction problems. The algorithm utilizes a level-wise tree construction strategy, each tree being incrementally developed layer by layer. In addition, it employs a histogram-based splitting technique, where histograms are constructed for each variable to determine the best variable split within each tree. This approach empowers the XGBoost algorithm to converge quicker, leading to faster prediction speeds. However, a downside of this method is that it may encounter inefficiencies when dealing with many missing values or data sets with unbalanced distributions. To counter the risk of overfitting, several regularization techniques and tree pruning methods are incorporated into the XGBoost algorithm. Additionally, the model provides a wide range of possible hyperparameters that can be fine-tuned for optimal performance during training. Furthermore, it offers parallel processing capabilities to effectively handle large databases.

- **LightGBM:** A more recent gradient-boosting algorithm, developed by Ke et al. [36], is LightGBM. Unlike XGBoost, LightGBM adopts a leaf-wise growth tree construction strategy, prioritizing nodes with the largest loss reduction rather than incrementally adding new layers. Furthermore, it employs the Gradient-Based One-Side Sampling (GOSS) technique for data-splitting. This technique selectively resamples data points and focuses solely on data instances with the greatest contribution to gradients. As a result, GOSS proves to be more efficient than XGBoost's histogram-based splitting approach, which computes gradients for all training instances. LightGBM also integrates an Exclusive Feature Bundling (EFB) method, which enhances runtime performance when dealing with many correlated variables. Overall, LightGBM's methodology focuses on optimizing computational speed while retaining the same accuracy levels, making it particularly beneficial for large datasets. However, compared to XGBoost, the LightGBM structure increases the risk of overfitting when only limited data is available. Similarly to XGBoost, LightGBM offers numerous regularization techniques and supports parallel training capabilities.

As mentioned above, the main differences between XGBoost and LightGBM relate to the approach of generating trees, which is visually represented in Figure 3. Anticipating which model is the superior one for this study is impractical. The model's performance may vary across scenarios and MPD events. Consequently, the performance of both models within the SPSO-CM framework is evaluated in Section 5.

Hyperparameter Tuning

As described, both models consist of numerous hyperparameters that can be adjusted to optimize performance. Grid searches are commonly employed to identify the optimal hyperparameter configuration. These searches involve testing various hyperparameter values to determine the ones that yield the best performance. Methods for conducting such searches include an exhaustive grid search of all possible combinations, a grid search that randomly selects different hyperparameter values, and a Bayesian search optimization technique.

In the Bayesian approach, promising hyperparameter values are predicted based on probability distributions. The algorithm first constructs a surrogate probability model ($p(y|hyperparameters)$) by tracking past evaluations. The next most promising hyperparameters are determined by optimizing this surrogate model. These chosen hyperparameters are then employed in the original objective function, leading to an updated version of the surrogate model. By iteratively repeating this process, the optimal set of hyperparameters is identified. In summary, the Bayesian Search algorithm spends a bit more time in intelligently selecting the next optimal hyperparameters to eventually decrease the number of potential hyperparameter combinations that should be evaluated. Consequently, the Bayesian approach proves to be more efficient than exhaustive and random grid searches, particularly when optimizing many hyperparameters. Due to these reasons, the Bayesian approach is utilized during the SPSO-CM's classification model's training phase. An overview of the tuned XGBoost and LightGBM hyperparameters is presented in Appendix B.

A cross-validation technique is required when fine-tuning the hyperparameters during training. Such techniques divide the complete training set into different training and test folds. The model's performance is assessed on each fold independently, leveraging all data points for optimal performance. Furthermore, applying cross-validation ensures that the initial test set is excluded during training and solely reserved to evaluate the model's performance on unseen data. In the context of the SPSO-CM framework, a nested group K-fold TimeSeriesplit cross-validation method is developed. The group K-fold technique initially splits the training dataset into four non-overlapping customer folds. Within each fold, the nested time-series method selects the most recent 40% of X_t as the evaluation set, while the remaining 60% of X_t are added to the other three folds as the training set. By repeating this process across all folds, four datasets containing the same data points but with varying divisions between training and testing sets are created. This implemented cross-validation technique ensures that all training data is used during fine-tuning, the relationship between X_t from the same customer is preserved, and possible data leakage is prevented.

Feature Selector

In addition to fine-tuning the hyperparameters of the selected gradient-boosting model, a feature selector is developed and incorporated into the Bayesian search space. It is essential to identify the importance of input features for prediction models as they heavily affect the model's performance. Irrelevant or marginally relevant input features can overcomplicate the model and negatively affect the model's performance. Generally, it is preferred to incorporate fewer features to enhance performance and reduce computational time.

The feature selector developed for SPSO-CM uses the Pearson correlation coefficient to assess the relationship between each input feature and the target value. The coefficient ranges from -1 to 1, with closer values indicating a stronger correlation, and 0 implying no correlation. After thorough testing of the feature subset and exclusion of heavily correlated features at the preprocessing stage, numerous input features might

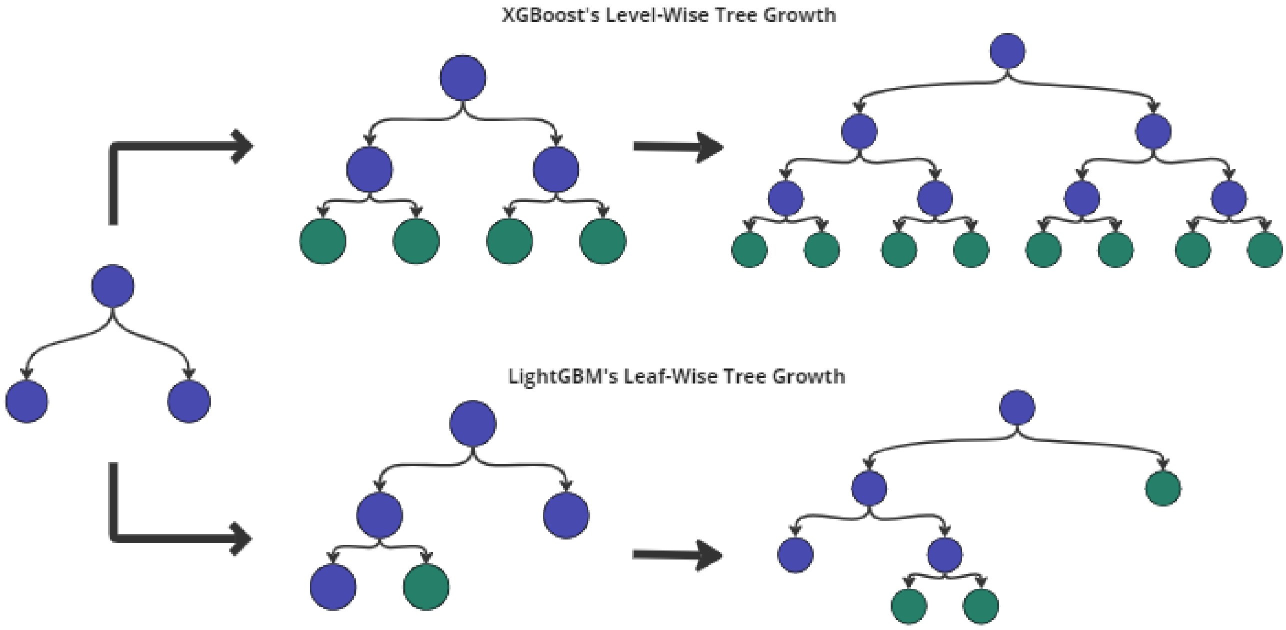


Figure 3: Differences between the XGBoost's level-wise tree construction method and LightGBM's leaf-wise tree construction method

remain relevant for certain PNs while inconsequential to others. During fine-tuning, the feature selector filters out uncorrelated features by employing distinct minimum correlation values in the Bayesian search algorithm's search space. As a result, the algorithm automatically selects the optimal feature sets for each prediction, leading to enhanced performance.

4.6 Evaluation Metrics

The final SPSO-CM layer consists of various error metrics to assess the model's performance. These metrics are widely utilized to evaluate binary classification models and interpret specific model characteristics. The main foundation of these error statistics is rooted in the confusion matrix, a performance measurement table that classifies the binary predicted values into four distinct categories. These classes are visually depicted in Figure 4 and are described below:

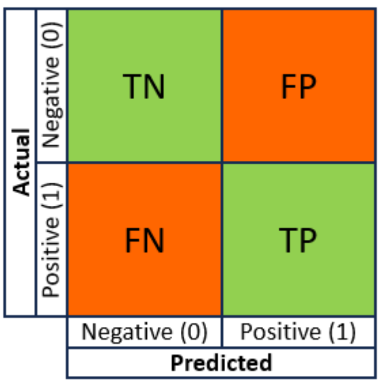


Figure 4: Visual representation of the confusion matrix

- **True Negative (TN):** Correctly predicts no subsequent PN purchase order occurrence.
- **True Positive (TP):** Correctly predicting a subsequent PN purchase order occurrence.
- **False Negative (FN):** Incorrectly predicting no subsequent PN purchase order occurrence, also known as a type II error.
- **False Positive (FP):** Incorrectly predicting a subsequent PN purchase order occurrence, also known as a type I error.

Most classification models primarily aim to maximize accuracy by minimizing both type I and type II errors. However, when dealing with class imbalance in classification models, relying solely on accuracy can be misleading. This is because high accuracy can be achieved by correctly predicting the majority class, without giving adequate attention to the minority class. To gain a more comprehensive understanding of model performance, it is essential to consider three other metrics: Precision, Recall and F1. These metrics focus on the positive class and can be calculated using the formulas presented in equations Equation 1, Equation 2, and Equation 3, respectively. Each metric examines different aspects of performance. *Precision* measures the ratio of accurately predicted positive instances to all predicted positives. *Recall*, on the other hand, computes the portion of correctly predicted positive instances out of all actual positives. In essence, a high precision score corresponds to a low type I error, whereas a high recall score indicates a low type II error. The *F1* score combines the precision and recall score in one error metric, balancing the two metrics.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

As discussed in this section, SPSO-CM can be configured in various ways. To determine the optimal configuration, the weighted average F1 score is calculated, as it is impractical to perform an SPSO-CM configuration evaluation for each PN individually. The weighted average F1 score takes into account the positive class distribution across all PNs, providing a comprehensive F1 score for the entire configured SPSO-CM. Once the best configuration is identified, an individual assessment of each PN is performed, as it is reasonable to assume that the prediction accuracy could vary among the included PNs, due to the limited availability of training instances or the presence of uncommon PN characteristics.

In addition to the individual calculation of the F1 score for each PN, two other metrics are used to assess the prediction performance of each subsequent PN purchase order. The first additional metric involves the examination of the precision-recall curve, which plots precision against recall scores, along with the calculation of its corresponding Area Under the Curve (AUC). This curve provides valuable information on the trade-off between precision and recall at different thresholds, making it more suitable to address class imbalance issues compared to the F1 score. A high AUC signifies strong performance in terms of both recall and precision, consequently reducing Type I and Type II errors. A no-skill classifier is represented by a horizontal line with a precision score of 0 and an AUC of 0, while a random classifier has an AUC of 0.5. Perfect classifiers achieve a turning point in the upper right corner of the graph, yielding an AUC of 1.

The second metric computed is the Matthews Correlation Coefficient (MCC). This metric considers the entire confusion matrix and is particularly useful when dealing with imbalanced datasets. MCC values range from -1 to 1, where -1 indicates poor performance, 0 suggests that the classifier performs no better than random chance, and 1 represents a perfect classifier [37]. The formula to compute the MCC score is provided in Equation 4.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FN) \cdot (TN+FN)}} \quad (4)$$

In an ideal scenario, all evaluation metrics should consistently demonstrate strong classifier performance. However, this is not always the reality, necessitating a thorough assessment of all metrics. Unfortunately, this complexity can pose challenges in interpretation, particularly for practitioners less familiar with SPSO-CM. To simplify interpretation and enhance clarity, it is decided to combine these three metrics into a single evaluation score. This combined performance indicator, named Model Effectiveness (ME), is calculated using the formula presented in Equation 5. Notably, the MCC score has been normalized to align its value range with that of F1 and AUC-PR. Furthermore, a higher weight has been assigned to the MCC, as it encompasses both positive and negative classes, while the F1 score and AUC-PR exclusively focus on the positive class.

$$ME = \frac{2 \cdot MCC_{nor} + F1 + AUC_{PR}}{4} \quad (5)$$

The ME score is bounded between 0 and 1. Consistent with the three included metrics, where 0 signifies a non-skill classifier, 0.5 represents a random classifier, and 1 signifies a perfect classifier, this study has established multiple thresholds to facilitate a more nuanced interpretation of performance:

- **Good Accuracy:** The SPSO-CM is considered highly accurate when it achieves a ME score greater than or equal to 0.7 ($ME \geq 0.7$) for predicting specific PN purchase orders. This threshold was chosen because an ME score of 0.7 implies that the SPSO-CM predictions significantly outperform random guessing, which corresponds to an ME score of 0.5, indicating a robust understanding of underlying data patterns.
- **Moderate Accuracy:** An ME score between 0.6 and 0.7 ($0.6 \leq ME < 0.7$) indicates moderate performance, surpassing random chance but still having a notable error margin.
- **Random Chance:** SPSO-CM's predictions fall within the range of 0.4 to 0.6 ($0.4 \leq ME < 0.6$), signifying performance equivalent to a random classifier.
- **Poor Accuracy:** An ME score below 0.4 ($ME < 0.4$) signifies poor performance, making SPSO-CM less effective than a random classifier. It is not recommended to use the SPSO-CM to predict subsequent PN purchase orders.

5 Results

This section evaluates the performance of the proposed SPSO-CM framework by considering two case studies, detailed in Subsection 5.1. The discussion begins with an examination of the generated features, particularly focusing on the derivation of the cluster features, in Subsection 5.2. Subsequently, Subsection 5.3 assesses various configurations of the SPSO-CM framework to determine the optimal setup. Following this, the SPSO-CM's individual PN prediction performance using the optimal configuration is discussed in Subsection 5.4. Lastly, the section concludes with a sensitivity analysis in Subsection 5.5, which describes the impact of some unique SPSO-CM components.

5.1 Case Studies

To assess the predictive capabilities of the SPSO-CM two distinct A320-family MPD events are simulated for a tw of 30, 60, and 90 days. The reason for two different MPD events is to test and evaluate the robustness of the model. A short overview of the two selected MPD events is summarized below:

- **MPD A:** This event is directly associated with one of the aircraft's water/waste airframe systems. It is recommended to be conducted at intervals of either 24 months or after accumulating 7500 flight hours. However, it is important to note that the initial interval could be adjusted based on the specific operators and the prevailing operating conditions. The execution of this event requires the availability of three first-level spare parts. Additionally, it requires an unconditional second-level spare part and a conditional third-level proprietary spare part, as indicated in the AMM.
- **MPD B:** MPD B presents a larger and more intricate scope compared to MPD A. It encompasses the comprehensive inspection of a specific flight control system of the aircraft. To have an airworthy aircraft, this event should not exceed an execution interval of 12,000 flight hours. The event entails a total of 23 spare parts, distributed over multiple AMM tasks: 6 second-level spare parts, 14 third-level spare parts, and three parts, which are referenced in both second-level and third-level AMM task descriptions.

A *tl* of 30 days was selected to generate the *adjacent set* and *adjacent set support* after discussions with company experts. Although longer periods were considered, this option was considered the most balanced between *tl* and different *tw*. Furthermore, the available company's sales records for simulating these two MPD events only included transactions between 2010 and 2020. To eliminate potential outliers caused by the disruptive impact of the COVID-19 pandemic on the aviation industry, it was chosen to exclude all transaction records from 2020 for further analysis. As a result, the sales database contained all X_t between 2010 and 2019.

5.2 Feature Results

As described in Subsection 4.3, the K-means clustering algorithm is employed to cluster PNs with similar characteristics, regardless of the associated MPD events. Through the use of the elbow method and the calculation of the silhouette score, the optimal cluster count was determined. A visual representation of the elbow method is shown in Figure 5.

The graph does not have a clearly distinguishable turning point. Consequently, the highest silhouette score, equal to 0.407 with 4 clusters, served as the decisive factor. Silhouette scores range from -1 to 1, where -1 implies poor separation, 0 suggests closely bound cluster boundaries, and 1 indicates optimal clusters. The relatively low score suggests that the resulting clusters lack distinctiveness, potentially due to the small dataset containing only 28 PNs, making cluster separation challenging. A visual representation of the differences between the four obtained clusters is illustrated in Figure 6, followed by a description of each cluster. The impact of the derived C_{PN} on the performance of SPSO-CM is discussed in Subsection 5.3.

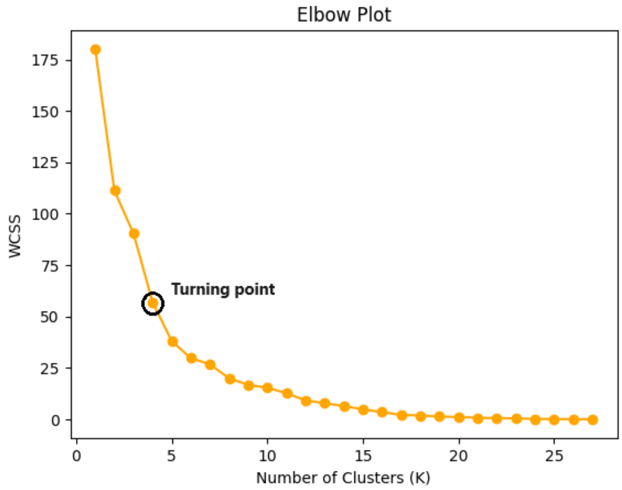


Figure 5: Visual representation of the Elbow Method to determine the optimal number of PN clusters. The turning point is supported by the highest silhouette score, equal to 0.407.

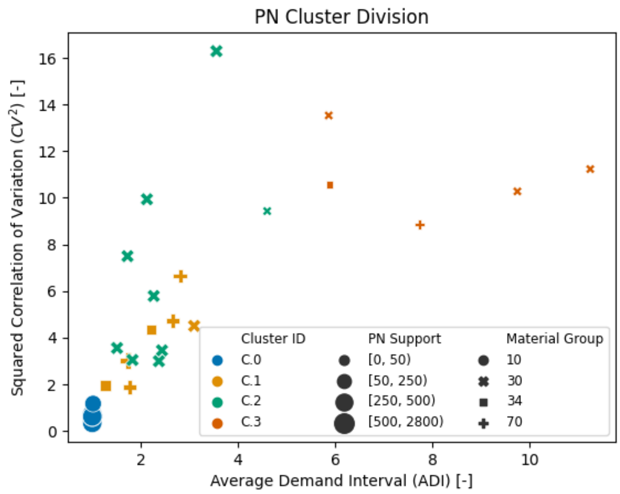


Figure 6: Visual representation of the four obtained clusters and their specific characteristics.

- **C.0:** This cluster includes all the proprietary PNs. The demand pattern is smooth or erratic with an ADI of approximately 1.0 and a CV^2 that ranges between 0.33 and 0.63. Furthermore, the PNs in this cluster are characterized by a high N_{PN} and are associated with a significant $N_{PN,customers}$.
- **C.1:** PNs in this cluster can be categorized as vendor or standard parts. They show a lumpy demand pattern with an average ADI of 2.22 and a CV^2 of 3.86. The N_{PN} within the sales records and the associated $N_{PN,customers}$ are at a medium level.
- **C.2:** This cluster includes vendor parts exhibiting a lumpy demand pattern similar to C.1, but with a greater variation in CV^2. Both N_{PN} and $N_{PN,customers}$ for these parts are at a low-medium level. Company experts have identified these parts as standard components that are often used in aircraft. However, the reason behind the

lumpy demand patterns, low-medium N_{PN}, and $N_{PN,customers}$ is that these parts are typically ordered in bulk quantities from specialized hardware suppliers.
- **C.3:** PNs in this cluster can be classified as standard or vendor parts and exhibit extremely lumpy demand behavior. The lowest ADI equals 5.86 and CV^2 starts from 8.84. Moreover, the N_{PN} and $N_{PN,customers}$ that have purchased these PNs are extremely low, indicating that these PNs are rare or purchased from another aftermarket distributor.

During SPSO-CM's evaluation, it became evident that most derived input features enhance accuracy, except for Wh_{X_t}, which had a negative effect on the model's prediction performances. Therefore, this feature is excluded from future analysis. Additionally, including Mat_{PN}, a cluster feature, as an additional order feature improved performance. This resulted in a total of 24 input features for each X_t associated with MPD A, including 17 lagged features (5 lagged features indicating each MPD-related PN value of ΔT_{PN}, $Qnt_{\Delta T_{PN}}$, $N_{X_{PN,t}}$), 6 order features (PN, $customer$, $priority$, $month$, $year$, Qnt_{X_t}), and 1 specific PN property feature (Mat_{PN}). For MPD B, the number of input features is higher due to the computation of lagged features for each of the 23 PNs, resulting in a total of 78 input features.

The evaluation also revealed that the highest accuracy across all error metrics was achieved by scaling all numerical features using the MinMax Scaler and encoding categorical features with the target encoding technique. Although decision trees do not necessitate numerical feature scaling, applying MinMax scaling led to improved accuracy. One plausible explanation for this lies in the diverse purchase behaviors of different customers, which result in varying lagged feature values. The MinMax scaler normalizes these values, thereby enhancing the interpretability of features for the gradient-boosting model. Additionally, using the target encoder, rather than the One Hot encoder, resulted in higher prediction accuracies. This is probably because the target encoder takes the target label into account when transforming categorical features, while the One Hot encoder increases feature dimensionality and introduces data sparsity.

5.3 SPSO-CM Performance

Various iterations of the SPSO-CM framework were examined to identify the most effective configuration that yields optimal performance indicators. These SPSO-CM configurations differ in their choice of gradient-boosting algorithm, inclusion of C_{PN} as an additional cluster feature, and the different X_t priorities selected within the data sample. This resulted in the evaluation of eight distinct SPSO-CM configurations across three different tw: 30 days, 60 days, and 90 days for each MPD event. The differences between configurations are denoted by their unique model configuration codes, referred to as *M-CF-DS*:

- *Model (M)*: Signifies the chosen gradient-boosting algorithm, which can be either XGBoost (X) or LightGBM (L).
- *Cluster Feature (CF)*: represents whether or not the cluster feature is included. The presence of the cluster feature is indicated with a C, while its absence is indicated by an N.
- *Data Subset (DS)*: Indicates which X_t priority types are included in the data set when using SPSO-CM. Two different data subsets were considered: a subset that contains all available X_t (AO) and another that contains only X_t instances with WSP or RTN priorities (SO), potentially excluding unplanned/unpredictable maintenance events typically indicated by AOG or USR priority codes.

For example, a configuration that utilizes an XGBoost gradient-boosting model with an additional cluster feature and is trained on all data priorities is denoted as *X-C-AO*, while a configuration that employs a LightGBM gradient-boosting algorithm without the cluster feature and is only trained on WSP and RTN purchase order priorities is denoted as *L-N-SO*.

To analyze the behavior of SPSO-CM across different time windows (tw), a distribution of weighted F1 scores is depicted for all possible configurations using multiple boxplots, shown in Figure 7.

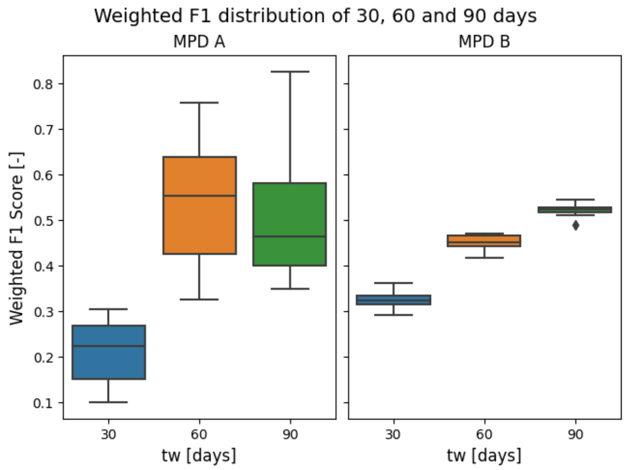


Figure 7: Comparison of the weighted F1 distribution for each tw, visually represented by separate boxplots.

The results reveal that all configurations perform poorly for a tw of 30 days, with medians of 0.22 for MPD A and 0.32 for MPD B. Consequently, these configurations are excluded from further analysis. On the contrary, the findings demonstrate an enhancement in SPSO-CM performance as tw increases. This can be explained as it becomes more likely that customers will place subsequent purchase orders when considering a longer tw.

For tw values of 60 and 90 days, the weighted F1 distributions for both MPD events show more promising results. Specifically, MPD B showed that longer future order windows are more predictable, as evidenced by the slightly higher position of the 90-day boxplot compared to the 60-day boxplot. Although this trend is less pronounced in the results of MPD A, a closer examination reveals that the maximum and minimum values of the 90-day boxplot surpass those of the 60-day boxplot. This suggests that a 90-day forecast is likely to be more accurate than a 60-day forecast, reinforcing the notion that longer future order windows are easier to predict.

Additionally, it is worth noting that SPSO-CM's configuration had a more significant impact on MPD A compared to MPD B, resulting in higher weighted average F1 scores for MPD A. This performance difference can be attributed to the complexity between the two events. MPD A consists of only 5 PNs, with 4 belonging to the same cluster, while MPD B includes 23 PNs distributed across all available clusters. This variation in the number and distribution of PNs can affect the weighted F1 score, as some PNs are more difficult to predict due to their unique characteristics.

Next, the influence of the two distinct *DS* on SPSO-CM prediction performance is assessed. Similarly to the previous analysis, the distributions of the weighted F1 scores are represented as boxplots in Figure 8. However, this time, only scores from the 60-day and 90-day *tw* experiments are included. The presented boxplots clearly demonstrate that the model performs better in predicting *SO* than in predicting *AO*, particularly concerning MPD A. This performance difference can be explained by the exclusion of unplanned X_t, which are inherently unpredictable. Furthermore, it is interesting to note that the weighted F1 scores for different *AO* configurations are relatively similar, except for a couple of outliers. On the contrary, the *SO* scores exhibit a wider spread, indicating that the selected configuration has a more significant impact on performance. Consequently, for the remainder of the analysis, the focus will only be on the *SO* configuration evaluated for the 90-day forecast.

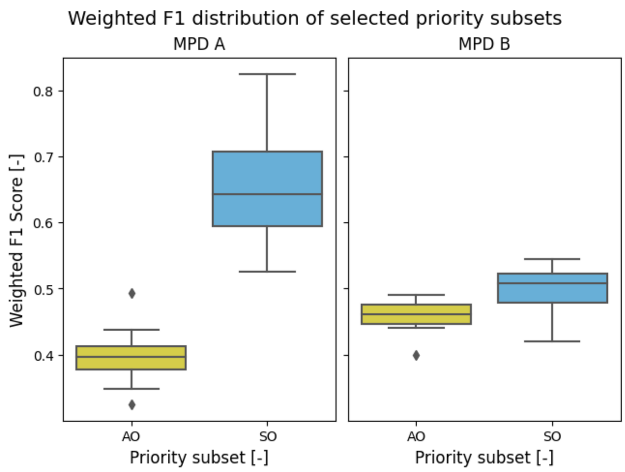


Figure 8: Comparison of the weighted F1 distribution for each *DS*-configuration, visually represented by separate boxplots.

The impact of the selected *M* is detailed in Table 1. When exclusively comparing the two gradient-boosting models while maintaining consistent configuration settings across the board, the results reveal that *X*-configurations consistently outperform *L*-configurations. As previously discussed in section 4.5, LightGBM models are specifically designed to enhance accuracy and optimize computational speed when handling large datasets. Therefore, one plausible explanation for the superior performance of the XGBoost model is that, for most PNs, there is a limited number of PN occurrences. Additionally, when more training data is available, LightGBM models can achieve higher weighted F1 scores.

Table 1: The weighted average metric scores for different *TS*-configurations forecasting a 90-day *tw*

	MPD A		**MPD B**	
Model	*F1* [-] ↑	*Runtime* [s] ↓	*F1* [-] ↑	*Runtime* [s] ↓
X-N-SO	**0.825**	317.05	0.524	1620.69
X-C-SO	0.690	327.31	**0.545**	1709.15
L-N-SO	0.525	395.89	0.522	1826.41
L-C-SO	0.543	389.89	0.518	2670.85

Finally, the different *CF* for the available *X-SO* configurations are evaluated. The results displayed in Table 1 present contradictory observations. MPD A's weighted F1 score decreases significantly from 0.825 to 0.690 when including C_{PN} as an input feature. In contrast, MPD B's performance slightly improves with the addition of C_{PN}, probably due to varying PNs and their diverse C_{PN} values. Incorporating C_{PN} as an input feature appears promising for predicting complex MPD events with various PNs, as it accounts for the properties of the PNs and their influence on future purchases.

In conclusion, the analysis clearly demonstrates that the *X-N-SO* configuration for MPD A and the *X-C-SO* configuration for MPD B outperform the other proposed configurations in terms of the weighted F1 score. Consequently, the selection for further analysis is based on these configurations. A concise summary of the *X-N-SO* configuration for MPD A and the *X-C-SO* configuration for MPD B can be found in Table 2 and Table 3, respectively. It is important to note that the SPSO-CM performs poorly for a 30-day *tw*, but its performance significantly improves with longer *tw*, leading to higher F1 scores. Additionally, there is a substantial difference in computational runtime between the two MPD events, which can be attributed to the larger training dataset, more PNs, and additional input features associated with MPD B.

5.4 Individual PN Performance

To better understand the performance of the optimally configured SPSO-CM, an in-depth analysis was performed by evaluating the predicted subsequent orders for each PN individually. The results of this analysis are visually displayed in a scatter plot shown in Figure 9.

Table 2: Results of the best performing SPSO-CM configuration, *X-N-SO*, for MPD A

tw [days]	Weighted Precision [-] ↑	Weighted Recall [-] ↑	Weighted F1 [-] ↑	Runtime [s] ↓
30	0.069	0.222	0.102	319.58
60	0.719	0.861	0.757	324.10
90	0.867	0.793	0.825	371.05

Table 3: Results of the best performing SPSO-CM configuration, *X-C-SO*, for MPD B

tw [days]	Weighted Precision [-] ↑	Weighted Recall [-] ↑	Weighted F1 [-] ↑	Runtime [s] ↓
30	0.339	0.305	0.292	1560.78
60	0.462	0.496	0.464	1590.80
90	0.547	0.576	0.545	1709.15

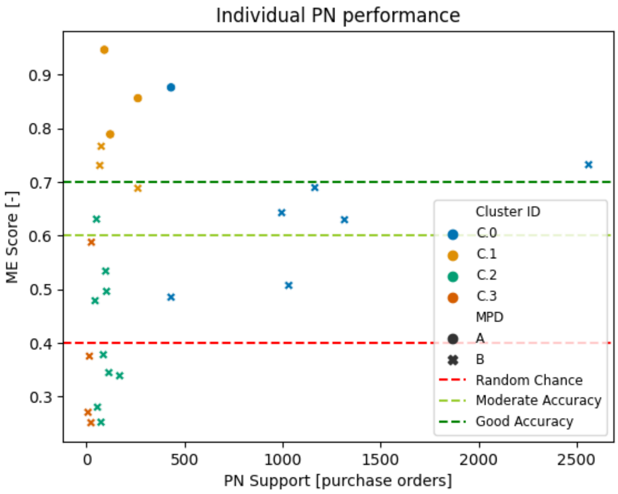


Figure 9: Graphical representation of the ME score against the PN support value to identify if a subsequent PN purchase order is predictable or not.

Upon examining the ME scores of all PNs as shown in Figure 9, two main observations emerge. First, there is a significant performance difference between PNs of MPD A and MPD B, aligning with the earlier comparison of the weighted F1 scores. As noted previously, this performance difference can be attributed to the exclusive presence of C.0 and C.1 PNs in MPD A, which inherently possess greater predictability, as illustrated in the graph. Furthermore, it is possible that there exists a more pronounced relationship among the PNs of MPD A compared to those of MPD B, possibly due to the inherent characteristics of MPD events.

The second observation that emerges is the clear pattern between the earlier derived clusters and the corresponding PNs' ME score. This pattern suggests a correlation between specific PN properties and the SPSO-CM's ability to predict subsequent PN purchase orders. The primary findings for each cluster are summarized below:

- **C.0:** This cluster exclusively consists of proprietary PNs, indicating that the company is the sole distributor of these PNs with no competition. As a result, more training data is available to predict purchase orders for these PNs and there is no interference from other distributors, leading to a more precise accuracy. The calculated ME scores confirm this, indicating that most of these PNs achieve moderate accuracy scores. However, two PNs are less predictable, exhibiting ME scores that suggest random behavior. In general, PNs that belong to this cluster can be considered predictable.
- **C.1:** Interestingly, PNs within this cluster achieve significantly higher ME scores compared to those in C.0. ME scores consistently meet or exceed the threshold for good accuracy. This suggests a well-balanced training dataset and a clear correlation between input features and PN purchase order occurrences. The SPSO-CM is exceptionally well suited for forecasting purchase orders for PNs in this cluster.
- **C.2:** ME scores indicate that PNs in this cluster are unpredictable, ranging from random to poor performance. It is not recommended to use the SPSO-CM to forecast these types of PNs.
- **C.3:** PNs that belong to this cluster exhibit an extremely lumpy demand behavior, which, according to the existing literature, is very difficult to predict. The results confirm this difficulty, as all ME scores for these PNs fall below the random chance threshold, indicating poor prediction precision. The low N_{PN} and the limited number $N_{PN,customers}$ already suggested that forecasting these purchase orders would be difficult. Consequently, the SPSO-CM should not be used to predict upcoming purchase orders associated with these PNs.

In summary, a direct correlation is observed between the inherent characteristics of PNs and the corresponding predictive capabilities of the SPSO-CM. The results in this section underscore the exclusive applicability of SPSO-CM to PNs belonging to C.0 or C.1, which typically include proprietary PNs or those with a significant number of available training instances. Moreover, noticeable variations in accuracy performance have been observed when comparing PNs sourced from MPD A and MPD B, suggesting that certain MPD events are better suited to predict subsequent PN purchase orders. Nevertheless, to comprehensively evaluate the model's performance, it is crucial to assess it over an extended time period and across numerous MPD events. Additionally, as more data on subsequent PN purchase orders become available, a more precise performance assessment can be achieved.

5.5 Sensitivity Analysis

The SPSO-CM incorporates several unique components and their impacts, when considering the X-N-SO configuration for predicting purchase orders associated with MPD A, are summarized in Table 4. Each component contributes positively to the predictive performance of the proposed algorithm.

Table 4: Ablation Study Results Compared to the Best Performed SPSO-CM: *X-I-SO*

Removed Component	ΔF1 [%]	ΔRuntime [%]
Resampler	-48.70	6.18
Feature Selector	-24.33	-1.15
Nested cross-validation	-31.00	-0.58

Removing the resampler from the proposed SPSO-CM results in a 48.7% decrease in the weighted F1 score. This aligns with existing literature, suggesting that data undersampling or oversampling improves machine-learning accuracy when dealing with class imbalance within datasets. Additionally, it demonstrates that undersampling maintains accurate predictions for most instances in the negative class, reflecting the inherent data characteristics. Furthermore, removing the resampler also leads to a slight decrease in runtime, which is expected since the model trains on fewer data instances.

Regarding the next distinctive component, the feature selector, its exclusion diminishes the SPSO-CM's predictive capabilities, reducing the weighted F1 score by 24.33%. This finding supports the hypothesis that various input features exert distinct influences on specific PNs, especially when considering the lagged input features that are unique to each PN. This suggests that some PNs within an MPD event have stronger associations with others, while some do not. Incorporating the feature selector naturally increases runtime due to the optimization of an additional hyperparameter. However, the runtime increase is minimal and can be disregarded. This may be attributed to situations in which fewer input features are utilized during the hyperparameter search, resulting in faster runtimes.

Lastly, the impact of the nested cross-validation is evaluated by comparing it with an original group K-fold cross-validation approach. The results, as listed in Table 4, demonstrate that the weighted F1 score is heavily influenced by the removal of the nested cross-validation component. In summary, incorporating nested cross-validation in the SPSO-CM not only increases the availability of training data during hyperparameter optimization but also mimics the original train-test split, where the same customers are present in both the training and testing sets. This approach provides the model with better insights into the relationships between customers and their corresponding X_t, making it a valuable addition to the SPSO-CM.

6 Discussion & Potential Applications

This section discusses the potential impact of the newly developed SPSO-CM on aftermarket distributors. As demonstrated in Section 5, SPSO-CM excels in producing accurate forecasts for proprietary PNs and PNs with a moderate amount of training data and a high-medium variety of $N_{PN,customers}$. However, when dealing with PNs characterized by extremely lumpy demand patterns or limited training data, SPSO-CM struggles to yield accurate results and often performs below average compared to a random classifier.

Before delving into the SPSO-CM's potential for aftermarket distributors, it is important to address some limitations and assumptions. Currently, the model is evaluated on a small dataset of 28 PNs across 2 MPD events. Drawing definitive conclusions about its performance requires further research involving a larger number of MPD events and a diverse set of PNs. Furthermore, the current framework assumes unique PNs for each MPD event, which is not realistic. Investigating the SPSO-CM's performance when PNs are shared among events would provide valuable insights. Another approach is to include all specific PN-related MPD events, but this would introduce a significant amount of lagged input features, leading to increased computational time and potential data sparsity.

Additionally, the SPSO-CM currently employs a gradient-boosting model. Future research could explore the use of neural networks, which are known for handling lumpy demand datasets effectively. This shift would also enable multiclass prediction possibilities, with each label representing potential purchase order occurrences for PNs in specific MPD events. However, it should be noted that this approach would result in longer computational times and increase the complexity of the SPSO-CM, making it less interpretable.

With these findings in mind, the practical implications of the proposed SPSO-CM algorithm are discussed both from a customer-level perspective in Subsection 6.1 and from a broader supply chain perspective in Subsection 6.2, while also considering the current limitations of SPSO-CM and potential areas for future research.

6.1 SPSO-CM for Customer-level Applications

The SPSO-CM, designed and trained to predict subsequent PN purchase orders at individual customer levels, is ideal for serving as a customer PN recommendation system. These systems, also known as *Other customers also bought* algorithms, are widely adopted in many e-commerce businesses. Numerous studies demonstrated their ability to increase sales and improve customer satisfaction, allowing the e-commerce business to increase product prices [38, 39]. In addition to the company-developed TDF algorithm, which identifies all correlated MPD PNs, the SPSO-CM can provide probabilities indicating a customer's likelihood of needing specific PNs to complete an MPD event. In this use case scenario, the ultimate decision to purchase the predicted PNs remains with the customer, mitigating any direct impact on the aftermarket distributor in the event of an inaccurate prediction. This additional service provides customers with essential information to make informed decisions about subsequent PN purchases. Consequently, this can lead to improved maintenance schedules and potentially reduce aircraft downtime, as all required PNs are readily available during the execution of a maintenance event.

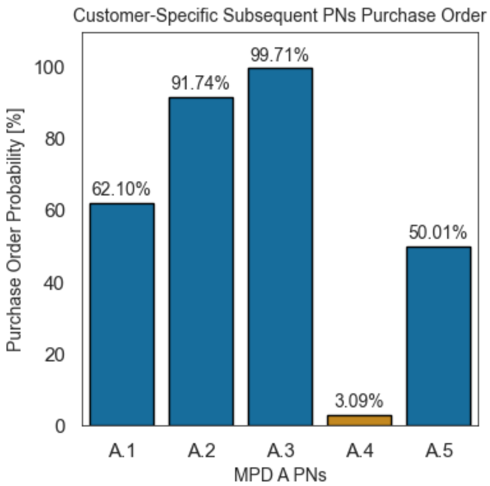


Figure 10: The SPSO-CM used to predict subsequent MPD A PN purchase orders for a specific customer that just purchased A.5

To illustrate the potential of SPSO-CM as a recommendation system for subsequent PN purchase orders, a real-case example featuring a customer who recently purchased PN A.5, is depicted in Figure 10. The probability forecast indicates a high likelihood that the customer will purchase A.2 and A.3 in the next 90 days, a slightly higher probability than 50% for A.1 and A.5, and a nearly zero probability for A.4. The customer can utilize this information by discussing it with technicians and warehouse personnel to determine whether to take action based on these probabilities and purchase the additional predicted PNs or safely disregard the forecast. Overall, this application helps aircraft carriers and MROs to generate a complete bill of material for a specific maintenance event.

A current limitation of the proposed SPSO-CM, and consequently the potential recommendation system, is its customer-specific nature. As highlighted in the literature review in Section 2, it was found that variations in customer buying behaviors have a significant impact on the accuracy of prediction models. Customers who frequently purchase PNs may exhibit greater predictability compared to those with irregular purchasing patterns. Therefore, more research is needed to understand how different customer profiles affect the accuracy of the SPSO-CM before providing customers with subsequent PN purchase order forecasts. A potential research approach could involve segmenting customers based on loyalty, purchasing behavior, company size, fleet type, and fleet age, and evaluating the SPSO-CM's performance within these segments. Additionally, exploring the potential value of including customer clusters as additional input features could provide valuable insights.

When the SPSO-CM is individually evaluated for multiple customers and tested across various MPD events, it opens up new business opportunities for aftermarket distributors. By accurately predicting upcoming purchase orders, aftermarket distributors can offer the SPSO-CM forecast as an additional service to customers. In addition to additional service revenue, this collaboration between an aftermarket distributor and their customers can enhance the SPSO-CM performance by integrating customers' maintenance schedules, component deterioration data, and current fleet status. Another promising business opportunity for aftermarket distributors is to proactively adjust or reduce PN prices based on these individual forecasts. For instance, when the SPSO-CM indicates a high probability of subsequent PN orders, distributors can propose bundled packages of multiple PNs at reduced prices, making it more attractive for customers to purchase all PNs together. Conversely, if customers postpone specific PN purchases despite the SPSO-CM predictions, the distributor may increase prices due to the increased complexity of on-time delivery, added shipping costs, and shifts in purchase order priorities. Furthermore, this proactive approach introduces the possibility of including unpredictable PNs from clusters C.2 and C.3 in the bundles when the TDF indicates a conditional relationship with the more reliable forecasted PNs from clusters C.0 and C.1, potentially leading to increased purchases of unpredictable PNs. In summary, this shift from a reactive business perspective to a proactive environment can increase sales, customer satisfaction, revenue, and even reduce the carbon footprint due to the combination of multiple shipments.

6.2 SPSO-CM for Supply Chain-level Applications

In addition to its potential applicability at the customer level, the SPSO-CM can be extended to a global scale to improve supply chain efficiency and optimize warehouse stock control. Instead of analyzing individual predictions, a monthly purchase forecast can be generated by aggregating the predicted PN purchase order occurrences ($\hat{y}$) over a month. The X_t are grouped per month, and for each month, the differences between the actual order occurrences (y) and the predicted occurrences ($\hat{y}$) are calculated. This calculation is called the purchase error and is equal to $FN - FP$. The purchase error indicates that achieving a perfect prediction of every purchase order occurrence may not be as crucial. This is because if there is an equal number of FN and FP on a monthly basis, the stock inventory level remains accurate. Thus, even with sub-optimal prediction performance, the SPSO-CM can be utilized for monthly PNs demand predictions.

To illustrate this, an example is given in Table 5, where the monthly purchase error is calculated for A.1. The results indicate that there is some variation within the purchase error across the four months. For instance, in July, the algorithm exhibited an error of -5, whereas in June, the error was 0. To provide a more comprehensive evaluation of this application, the mean absolute error (MAE), the root mean squared error (RMSE), and the mean absolute percentage error (MAPE) were calculated. For formulas and additional information on these metrics, the interested reader may refer to [40]. The results of these calculations for A.1 are detailed in Table 6.

Table 5: The purchase error of A.1 per monthly X_t.

Month	TN	TP	FP	FN	Purchase Error
June	4	3	1	1	0
July	2	4	5	0	-5
August	3	4	2	4	2
September	3	4	4	1	-3

Table 6: Error metrics derived from the monthly purchase errors of A.1

	MAE ↓	RMSE ↓	RMSE [%] ↓
A.1	2.50	3.08	21

The calculated error metrics for A.1, with MAE and RMSE of 2.50 and 3.08 for misclassified purchase orders, respectively, fall within an acceptable range. Especially when considering an average of 11.25 purchase predictions per month. This reaffirms that in supply chain applications, the focus is on maintaining a balance between FN and FP on a monthly basis to ensure accurate stock inventory levels, rather than achieving perfect predictions for every PN purchase order occurrence. It is important to note that the example above does not exactly match a real-case scenario. In this case, the grouping of predicted PN purchase order occurrences into monthly intervals is based on the initial X_t purchase date. Given a 90-day period tw, there is a significant probability that subsequent predicted PN purchase orders may occur in a different month. Nonetheless, it is assumed that the results of this preliminary analysis still provide a valuable indication of its intended purpose for future research or implementation.

Expanding on the performance findings of the SPSO-CM in supply chain applications, the creation of a purchase order probability forecast becomes possible. This SPSO-CM application has the potential to improve the current demand prediction algorithms used by the aftermarket distributor, ultimately leading to more effective stock control. In order to develop this, its is important to understand that the SPSO-CM essentially evaluates the probability that purchase orders occur based on probability values. When the predicted probability is below 50%, it signifies a low likelihood of a subsequent PN purchase order and predicts that it will not occur. Conversely, a probability higher than 50% suggests a higher likelihood of an order and predicts its occurrence. Figure 11 visually represents the probability distribution for the predicted purchase orders of A.1 when considering the test dataset. Using these probabilities, a forecast can be generated to estimate the probability of a specific number of purchase orders for the next 90 days, considering all unfulfilled predictions of X_t made in the last 90 days. An illustrative example of this application for A.1 can be found in Figure 12. The formula for calculating the probability of K purchase orders out of N predicted purchase orders is presented in Equation 6, where p stands for the predicted SPSO-CM order occurrence probability, N for the total events (total available probabilities) and K for the target number of purchase order occurrences.

$$P(K) = \sum_{1 \leq i_1 < i_2 < \ldots < i_K \leq N} \left(\prod_{j=1}^{K} p_{i_j} \right) \left(\prod_{j=K+1}^{N} (1 - p_{i_j}) \right) \tag{6}$$

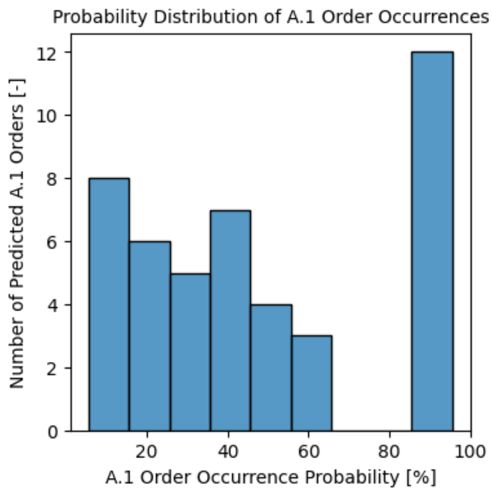


Figure 11: Probability distribution of A.1 purchase order occurrences when prediction the initial X_t within the test dataset

The depicted probability forecast indicates a higher probability of 6 or 7 purchases of A.1 in the first few weeks. As time progresses and the number of considered X_t decreases, the maximum number of expected purchase orders also decreases. In summary, the highest probability line should provide a reliable estimate of the maximum number of subsequent PN purchase orders expected during that period. Furthermore, the forecast should be updated weekly by including the most recent X_t to have a comprehensive probability forecast.

As already described, this probability forecast application can positively contribute to existing demand prediction models, resulting in a more effective optimization of warehouse inventory levels for specific PNs. By forecasting potential subsequent purchase orders for

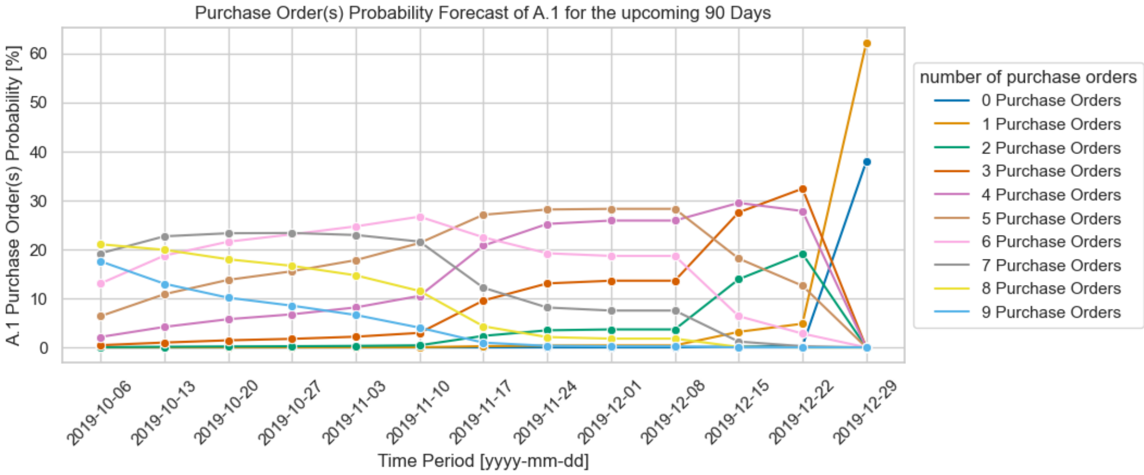


Figure 12: SPSO-CM used to forecast the probability of subsequent A.1 purchase orders for the upcoming 90 days

these PNs, inventory management becomes more precise. This precision enables aftermarket distributors to proactively address potential stock shortages, leading to increased customer satisfaction and potentially higher sales volumes. Additionally, it helps in identifying frequently sold PNs and those with lower demand, simplifying the process of removing less promising products to create space for others.

However, before implementing the purchase order probability forecast in practice, a thorough assessment of its accuracy is needed in real-world scenarios over an extended period. Additionally, the SPSO-CM currently focuses exclusively on predicting the occurrence of subsequent PN purchase orders. An interesting direction for future research is to extend the existing SPSO-CM framework with an associated subsequent PN purchase order quantity forecast. This expansion would require either creating a new prediction model or modifying the existing SPSO-CM by shifting its training objective from classification to regression, with target labels representing the respective purchase order quantities rather than binary occurrences. Developing this capability would provide even greater value for the aftermarket distributor.

7 Conclusion

This paper presents a novel classification model (SPSO-CM) to predict subsequent customer-specific purchases of maintenance planning document (MPD) related spare parts based on technical documentation and previous sales records. The model is utilized to forecast spare part purchase order occurrences within 30, 60, and 90-day time windows for an aircraft aftermarket distributor.

The proposed model's architecture centers around a gradient-boosting algorithm and integrates several unique components to enhance its performance. To begin, it employs a feature creator that generates input features by analyzing previous purchases of related spare parts, the initial purchase order details, and spare part characteristics. A k-means clustering algorithm is used to group spare parts with similar characteristics, as certain specific spare part properties significantly influence demand prediction models. Four primary clusters were identified, ranging from a segment of only proprietary spare parts to a group of spare parts that exhibits an extreme lumpy demand pattern.

Regarding the gradient-boosting algorithm, the accuracy performance of both the XGBoost and LightGBM algorithms was evaluated within the framework. It was determined that XGBoost outperformed the LightGBM model in terms of computational time and F1 scores, making it the preferred gradient-boosting algorithm for further analysis. Furthermore, to address class imbalances between the occurrence and non-occurrence of purchase orders within the training dataset, the architecture incorporates a version-3 Nearmiss undersampling strategy. A unique feature selector and nested cross-validation technique were developed and incorporated into the Bayesian search space to optimize hyperparameters and improve performance.

Evaluating the SPSO-CM on 28 different spare parts across 2 MPD events provided valuable insight into its performance. The results highlighted a correlation between the derived clusters and the prediction accuracy of the proposed model. Notably, proprietary spare parts and those with a moderate purchase history from various customers demonstrated strong performance, while spare parts with lumpy demand patterns and limited available data exhibited poor accuracy scores. However, to draw a definitive conclusion, the SPSO-CM should be evaluated over an extended time period, across numerous MPD events and a more diverse set of spare parts.

When evaluating the SPSO-CM as a potential basis for applications, it demonstrates a significant potential value for aftermarket distributors. Moreover, it has the potential to introduce a new business perspective, transforming the current reactive environment into a more proactive approach. Two potential applications are discussed, one from a customer-level perspective and another at a larger supply-chain level, highlighting its promising capabilities. Future research should involve simulations to evaluate its true value in terms of revenue, stock efficiency, and customer satisfaction. Additionally, an interesting new research direction could involve the development of a separate model to predict not only purchase order occurrences but also the associated purchase order quantities. Such an approach would lead to a more advanced estimation of upcoming purchase orders, thereby enhancing supply chain efficiency and inventory management. Lastly, it would be of interest to explore the use of neural networks as the used prediction model instead of gradient-boosting algorithms, given their proficiency in handling complex demand patterns. However, it is essential to note that this choice may introduce complexity and longer computational times, which may not be desirable.

References

[1] Stan Shparberg and Bob Lange. Airbus Global Market Forecast 2022-2041 Presentation, 2022.

[2] Adel A. Ghobbar and Chris H. Friend. Evaluation of forecasting methods for intermittent parts demand in the field of aviation: A predictive model. *Computers and Operations Research*, 30(14):2097–2114, 2003. ISSN 03050548. doi: 10.1016/S0305-0548(02)00125-9.

[3] Xiaoqian Sun, Sebastian Wandelt, and Anming Zhang. How did COVID-19 impact air transportation? A first peek through the lens of complex networks. *Journal of Air Transport Management*, 89:101928, 10 2020. ISSN 0969-6997. doi: 10.1016/J.JAIRTRAMAN.2020.101928.

[4] Lolke van der Heide. Airbus kan de productie niet aan, 11 2022. URL `https://www.luchtvaartnieuws.nl/nieuws/categorie/18/technologie/airbus-kan-de-productie-niet-aan`.

[5] Jingyao Gu, Guoqing Zhang, and Kevin W. Li. Efficient aircraft spare parts inventory management under demand uncertainty. *Journal of Air Transport Management*, 42:101–109, 1 2015. ISSN 0969-6997. doi: 10.1016/J.JAIRTRAMAN.2014.09.006.

[6] MBA Aviation. Utilizing Alternative Collateral: Spare Parts. URL `https://www.mba.aero/utilizing-alternative-collateral-spare-parts/#:~:text=Rotable%20items%20are%20parts%20that,gear%2C%20and%20major%20engine%20accessories`.

[7] A. A. Syntetos, J. E. Boylan, and J. D. Croston. On the categorization of demand patterns. *Journal of the Operational Research Society*, 56 (5):495–503, 8 2005. ISSN 01605682. doi: 10.1057/PALGRAVE.JORS.2601841/TABLES/2. URL `https://link.springer.com/article/10.1057/palgrave.jors.2601841`.

[8] J. D. Croston. Forecasting and Stock Control for Intermittent Demands. *Journal of the Operational Research Society 1972 23:3*, 23(3):289–303, 9 1972. ISSN 1476-9360. doi: 10.1057/JORS.1972.50.

[9] Alberto Regattieri, M. Gamberi, R. Gamberini, and R. Manzini. Managing lumpy demand for aircraft spare parts. *Journal of Air Transport Management*, 11(6):426–431, 11 2005. ISSN 09696997. doi: 10.1016/J.JAIRTRAMAN.2005.06.003.

[10] A. H.C. Eaves and B. G. Kingsman. Forecasting for the ordering and stock-holding of spare parts. *https://doi.org/10.1057/palgrave.jors.2601697*, 55(4):431–437, 2017. ISSN 14769360. doi: 10.1057/PALGRAVE.JORS.2601697.

[11] Aris A. Syntetos and John E. Boylan. The accuracy of intermittent demand estimates. *International Journal of Forecasting*, 21(2):303–314, 4 2005. ISSN 01692070. doi: 10.1016/j.ijforecast.2004.10.001.

[12] Çerağ Pinçe, Laura Turrini, and Joern Meissner. Intermittent demand forecasting for spare parts: A Critical review. *Omega (United Kingdom)*, 105, 12 2021. ISSN 03050483. doi: 10.1016/J.OMEGA.2021.102513.

[13] Rafael S. Gutierrez, Adriano O. Solis, and Somnath Mukhopadhyay. Lumpy demand forecasting using neural networks. *International Journal of Production Economics*, 111(2):409–420, 2 2008. ISSN 0925-5273. doi: 10.1016/J.IJPE.2007.01.007.

[14] M. Z. Babai, A. Tsadiras, and C. Papadopoulos. On the empirical performance of some new neural network methods for forecasting intermittent demand. *IMA Journal of Management Mathematics*, 31:281–305, 2020. doi: 10.1093/imaman/dpaa003.

[15] K. Nemati Amirkolaii, A. Baboli, M. K. Shahzad, and R. Tonadre. Demand Forecasting for Irregular Demands in Business Aircraft Spare Parts Supply Chains by using Artificial Intelligence (AI). 50 (1):15221–15226, 7 2017. ISSN 24058963. doi: 10.1016/J.IFACOL.2017.08.2371.

[16] Daniel Graupe. *Principles Of Artificial Neural Networks*. 3 edition, 2013. ISBN 9789814522748.

[17] Josef Hochreiter. *Untersuchungen zu dynamischen neuronalen Netzen*. PhD thesis, Technische Universitat Munchen, Munchen, 6 1991.

[18] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[19] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8):5929–5955, 12 2020. ISSN 15737462. doi: 10.1007/S10462-020-09838-1/TABLES/1.

[20] Kiran Kumar Chandriah and Raghavendra V. Naraganahalli. RNN / LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting. *Multimedia Tools and Applications*, 80(17):26145–26159, 7 2021. ISSN 15737721. doi: 10.1007/S11042-021-10913-0/FIGURES/5.

[21] Stephan Chang and Felipe Meneguzzi. Forecasting Demand with Limited Information Using Gradient Tree Boosting. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference*, pages 227–230. Association for the Advancement of Artificial Intelligence, 2017. URL `https://cdn.aaai.org/ocs/15508/15508-68651-1-PB.pdf`.

[22] Shu Ching Wang, Yao Te Tsai, and Yi Syuan Ciou. A hybrid big data analytical approach for analyzing customer patterns through an integrated supply chain network. *Journal of Industrial Information Integration*, 20:100177, 12 2020. ISSN 2452-414X. doi: 10.1016/J.JII.2020.100177.

[23] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *1993 ACM SIGMOID international conference on Management of data*, pages 207–216. Association for Computing Machinery (ACM), 1993. doi: 10.1145/170035.170072.

[24] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In *Very large data bases*, pages 487–499, 1994. URL `https://www.vldb.org/conf/1994/P487.PDF`.

[25] Micheline Kamber, Jiawei Han, and Jenny Y Chiang. Metarule-Guided Mining of Multi-Dimensional Association Rules Using Data Cubes. 1997. URL `www.aaai.org`.

[26] Avadh Kishor Singh, Ajeet Kumar, and Ashish K. Maurya. An empirical analysis and comparison of apriori and FP-growth algorithm for frequent pattern mining. *Proceedings of 2014 IEEE International Conference on Advanced Communication, Control and Computing Technologies, ICACCCT 2014*, pages 1599–1602, 1 2014. doi: 10.1109/ICACCCT.2014.7019377.

[27] M Chen and H Wu. An association-based clustering approach to order batching considering customer demand patterns. *Omega*, 33(4):333–343, 8 2005. ISSN 03050483. doi: 10.1016/j.omega.2004.05.003.

[28] U. C. Moharana and S. P. Sarmah. Determination of optimal order-up to level quantities for dependent spare parts using data mining. *Computers & Industrial Engineering*, 95:27–40, 5 2016. ISSN 0360-8352. doi: 10.1016/J.CIE.2016.01.024.

[29] Matteo Kalchschmidt, Roberto Verganti, and Giulio Zotteri. Forecasting demand from heterogeneous customers. *International Journal of Operations & Production Management*, 26(6):619–638, 2006. doi: 10.1108/01443570610666975.

[30] Federico Caniato, Matteo Kalchschmidt, Stefano Ronchi, Roberto Verganti, Giulio Zotteri, Federico Caniatoy, Matteo Kalchschmidtz, Stefano Ronchiy, and Roberto Vergantiy. Clustering customers to forecast demand, Production Planning & Control. *Production Planning & Control*, 16(1):32–43, 2005. ISSN 1366-5871. doi: 10.1080/09537280512331325155.

[31] Harry A. Kinnison and Tariq Siddiqui. *Aviation Maintenance Management*, volume 2. Second edition, 2013. ISBN 978-0-07-180502-5.

[32] Hülya Güçdemir and Hasan Selim. Integrating multi-criteria decision making and clustering for business customer segmentation. *Industrial Management & Data Systems*, 115(6):1022–1040, 2015. doi: 10.1108/IMDS-01-2015-0027.

[33] Joos Korstanje. *Advanced forecasting with python: With state-of-the-art-models including LSTMs, Facebook's prophet, and Amazon's DeepAR*. Apress Media LLC, 7 2021. ISBN 9781484271506. doi: 10.1007/978-1-4842-7150-6.

[34] Candice Bentéjac, Anna Csörgő, and Gonzalo Martínez-Muñoz. A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3):1937–1967, 3 2021. ISSN 15737462. doi: 10.1007/S10462-020-09896-5/TABLES/12. URL `https://link-springer-com.tudelft.idm.oclc.org/article/10.1007/s10462-020-09896-5`.

[35] Tianqi Chen and Carlos Guestrin. XGBoost: A Scalable Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, New York, NY, USA, 8 2016. ACM. ISBN 9781450342322. doi: 10.1145/2939672.2939785.

[36] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. URL `https://github.com/Microsoft/LightGBM`.

[37] Qiuming Zhu. On the performance of Matthews correlation coefficient (MCC) for imbalanced dataset. *Pattern Recognition Letters*, 136:71–80, 8 2020. ISSN 0167-8655. doi: 10.1016/J.PATREC.2020.03.030.

[38] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of Recommendation Algorithms for E-Commerce. In *Proceedings of the 2nd ACM Conference on Electronic Commerce*, pages 158–167, Minneapolis, 10 2000. Department of Computer Science and Engineering - University of Minnesota.

[39] Bhavik Pathak, Robert Garfinkel, Ram Gopal, Rajkumar Venkatesan, and Fang Yin. Empirical Analysis of the Impact of Recommender Systems on Sales. *https://doi-org.tudelft.idm.oclc.org/10.2753/MIS0742-1222270205*, 27(2):159–188, 10 2014. ISSN 07421222. doi: 10.2753/MIS0742-1222270205. URL `https://www-tandfonline-com.tudelft.idm.oclc.org/doi/abs/10.2753/MIS0742-1222270205`.

[40] Nicolas Vandeput. Forecast KPIs: RMSE, MAE, MAPE & Bias, 7 2019. URL `https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d`.

Appendices

A Nomenclature

SPSO-CM Symbols

ΔT_{PN}	Time interval in days since the last purchase order of the specified PN.
$\overline{Qnt}_{PN}$	The average amount of PN sold per purchase order.
AO	SPSO-CM configuration that considers all priority types
$adjacentset$	Adjacent set of previous purchase orders within the predefined tl
C_{PN}	Unique cluster ID
$customer$	The customer of the purchase order line.
DS	Data Subset: Specifies which priorities types are presented for a SPSO-CM configuration.
$F_{adjacent}$	Derived support / frequency of the adjacent set.
M	Model: Specifies the SPSO-CM gradient-boosting configuration.
Mat_{PN}	PN material group
$month$	The month (date) of X_t.
$N_{PN,customers}$	The number of unique customers who purchased the PN.
N_{PN}	Total number of purchase orders associated with the specified PN.
$N_{PN-inter}$	Number of identified interchangeable items related to the respective PN.
$N_{X_{PN,t}}$	The total number of PN purchase orders until the most recent PN purchase order.
$priority$	The priority code of X_t.
$Qnt_{\Delta T_{PN}}$	The quantity of the most recent PN purchase order.
Qnt_{X_t}	The specified quantity of X_t.
SO	SPSO-CM configuration that only considers WSP and RTN purchase orders.
tl	time lag
CF	Cluster Feature: Specifies whether the cluster feature is part of the SPSO-CM configuration.
tw	time window
Wh_{X_t}	The warehouse of X_t.
$X_{PN,t-1}$	The previous PN purchase order by the same customer
X_t	purchase order line
$y_{PN,t+tw}$	The target label indicating the presence or absence of a subsequent purchase of PN in the upcoming tw.
$year$	The year (date) of X_t.

Abbreviations

ADI	Average Demand Interval
AMM	Aircraft Maintenance Manual
ANN	Artificial Neural Networks
ARM	Association Rule Mining
AUC	Aera Under the Curve
AUC_{PR}	Aera Under the Precision-Recall Curve
BP	Back Propagation
CD	Calendar Days
CSV	Comma Separated Values
CV^2	Squared correlation of variation
EFB	Exclusive Feature Bundling
FC	Flight Cycles
FH	Flight Hours
FN	False Negative confusion matrix class
FP	False Positive confusion matrix class
FPR	False Positive Rate
$GOSS$	Gradient-based One-Side Sampling
$GRMSE$	Geometric Root Mean Square Error
$LightGBM$	Light Gradient Boosting Model
$LSTM$	Longs-Short Memory Network
MAD/A	Mean Absolute Deviation Average
MAE	Mean Absolute Error
$MAPE$	Mean Absolute Percentage Error
$MASE$	Mean Absolute Square Error
MCC	Matthew Correlation Coefficient
MLP	Multi-layer Perceptron

MPD	Maintenance Planning Document
MRO	Maintenance Repair & Overhaul
OEM	Original Aircraft Manufacturer
PN	Spare Part ID
$RMSE$	Root Mean Squared Error
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SBA	Syntetos Boylan Approximation
SES	Single Exponential Smoothing model
$SPSO-CM$	Spare Part Subsequent Purchase Occurrence Classification Model
TDF	Technical Documentation Forecast
TN	True Negative confusion matrix class
TP	True Positive confusion matrix class
$WCSS$	Sum of squared distances to cluster centroids
$XGBoost$	Extreme Gradient Boosting model

B Gradient Boosting Hyperparameters

An overview of the used hyperparameters during the Bayesian search space is given below. The standard hyperparameters utilized for both XGBoost and LightGBM models.

Standard Hyperparameters

- **learning rate** $[0, 1]$**:** Controls the step size shrinkage at each boosting iteration, serving to make the model more conservative.
- **Maximum Depth** $[0, \infty]$**:** Represents the maximum depth of the tree and is used to prevent the model from overfitting. A higher maximum tree depth enables the model to learn highly specific relationships unique to individual data samples.
- **reg alpha:** This parameter governs L1 regularization, often referred to as Lasso regularization. L1 regularization introduces a penalty factor into the model's loss function, pushing it to diminish the influence of less important features by gradually reducing their weights to zero. As this parameter is increased, the strength of L1 regularization grows, resulting in a more pronounced feature selection process. Ultimately, it serves to combat overfitting, boost the model's generalization capacity, and improve its performance when faced with noisy or interrelated features.
- **reg lambda:** This parameter controls L2 regularization, also known as Ridge regularization. Unlike L1 regularization, it reduces the squared magnitude of the weights rather than the absolute values. It is used to mitigate overfitting.
- **subsample** $(0, 1]$**:** Indicates the ratio used to randomly sample the training instances before tree growth. Lower values are used to mitigate overfitting, but excessively low values can result in underfitting.
- **colsample bytree** $(0, 1]$**:** Dictates the subsample ratio to control the columns used at each level of the tree. It is applied once for every constructed tree.
- **colsample bynode** $(0, 1]$**:** Specifies the fraction of columns that are randomly chosen to split each node in a tree. It is applied individually to the current node that is split, not to the entire tree or level.
- **scale pos weight** $[1, \infty]$**:** Controls the balance between the positive and negative classes, particularly valuable when handling imbalanced datasets. Higher values indicate a bigger imbalance between the classes of the dataset.

Specific XGBoost Hyperparameters

- **gamma** $[0, \infty]$**:** Denotes the minimum required loss reduction to make a further partition on a leaf node. A larger gamma results in a more conservative algorithm.
- **colsample bylevel** $(0, 1]$**:** specifies the subsample ratio of columns used at each tree depth level.

Specific LightGBM Hyperparameters

- **num leaves:** Specifies the maximum number of leaves per tree. The more leaves the more conservative the algorithm becomes.
- **max bin:** This parameter sets the maximum number of discrete bins for bucketing input features. It improves training speed and reduces memory usage by reducing the number of evaluated splits.

II

Literature Study
previously graded under AE4020

1
Introduction

The aviation industry is a highly complex, competitive market with various stakeholders. Millions of travelers are traveling every day with thousands of competing airlines that are using different kinds of aircraft types. Over the last few decades, the entire sector grew steadily till the outbreak of the COVID-19 pandemic at the beginning of 2020, causing a massive market disruption affecting the entire transportation industry. Although the world is still recovering from the recent crisis, it is expected that air traffic is fully recovered in 2024 with respect to the 2019 levels. According to Airbus's Global Service Forecast [22], it is even expected that passenger traffic and freight traffic increases annually by 3.6% and 3.2%, respectively towards 2040. Considering this growth and the sustainability targets set by industry leaders [23], this will mean a total demand of 39,500 new aircraft in the upcoming 20 years. Resulting in an operational fleet of 46,930 aircraft compared to 22,880 aircraft in service in 2022 (pre-COVID times). This increase directly impacts aircraft maintenance organizations and the demand for spare parts, in general [7].

Airliners try to plan their maintenance events as efficiently as possible to increase their operational usage of an aircraft, which directly reduces their financial operating costs. It is estimated that approximately 10%-15% of the direct airline's operating costs are spent on keeping the aircraft maintained [24]. This only includes the Maintenance, Repair, and Overhaul (MRO) costs and does not even consider the additional costs that will apply during an unscheduled maintenance event.

Accurate forecasting of spare parts is one of the biggest challenges in the aviation industry, as their unavailability can lead to high downtime costs. Due to the high variation in demand and its unpredictability, this is a difficult process. A recent news article on the current spare parts delivery issue, caused by problems in the supply chain of Boeing and Airbus [25], illustrates these challenges. Most forecasting programs are based on the regulations such as maintenance schedules and usage patterns of parts or previous demand [20]. Aircraft operators can utilize this information for their prediction models. Aftermarket solutions, on the other hand, have limited access to these data sources. Their forecasting models are mainly based on previous purchases. Therefore, **The aim of this study is to provide an overview of spare part demand forecasting techniques that are applicable to the aviation aftermarket industry. Furthermore, research on improvement strategies to overcome the lacking data availability should also be conducted.**

This report, in order to achieve the research aim, is structured as follows. Starting with a general introduction to aircraft maintenance and its key stakeholders in Chapter 2. Followed by Chapter 3, where the main spare part demand characteristics are introduced, along with a detailed analysis of various forecasting techniques, evaluating their performance, structure, advantages, and disadvantages. The subsequent chapters focus on forecasting improvement strategies, exploring customer segmentation in Chapter 4, and a detailed explanation of two pattern mining techniques, association rule mining, and sequential pattern mining, in Chapter 5. This research concludes with an overview of the main findings, derived research questions, and feature scope of the project in Chapter 6.

2

Aircraft Maintenance

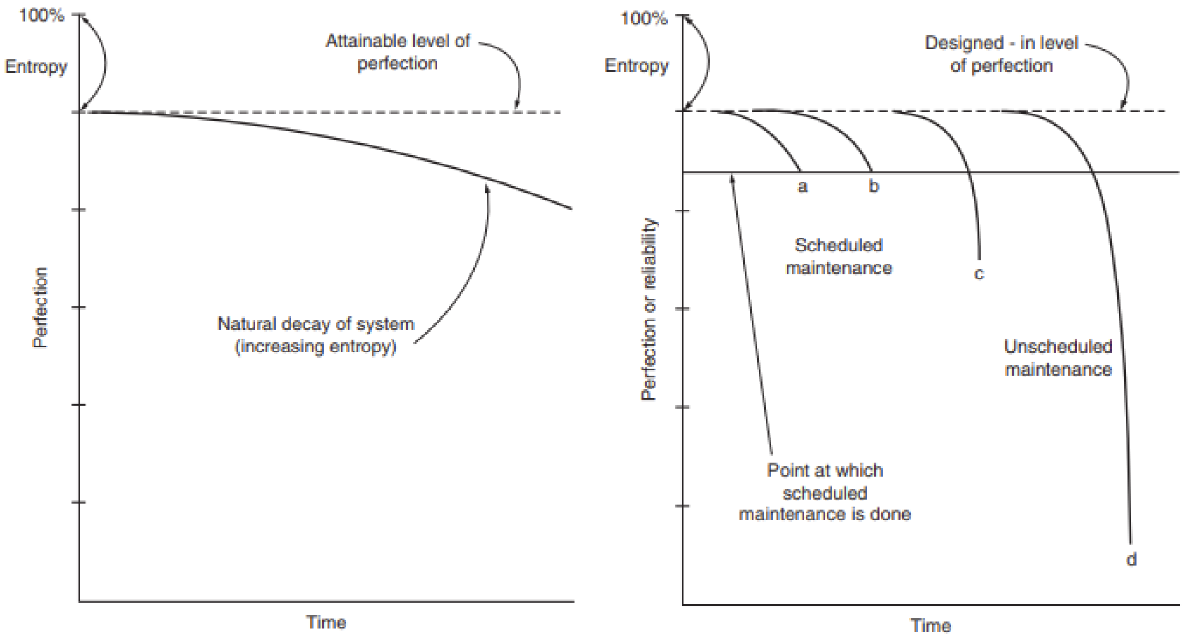
Aircraft consists of thousands of parts that must be maintained regularly to guarantee their operational state. Maintenance processes are complex and influenced by several factors, including the type of maintenance events and the characteristics of spare parts, which are discussed in detail in this chapter. An overview of aircraft maintenance processes and failure types is provided in Section 2.1. After which, multiple spare part characteristics are described in Section 2.2. The domain of aircraft maintenance involves multiple stakeholders, and an overview of they key parties can be found in Section 2.3. To conclude, a summary of all findings is presented in Section 2.4.

2.1. Aircraft Maintenance Characteristics

Aircraft maintenance activities are highly-regulated and listed by the aircraft manufacturer in a Maintenance Planning Document (MPD). This document consists of the Maintenance Review Board Report (MRBR) with additional suggested tasks by the aircraft's Original Equipment Manufacturer (OEM) and is unique for every operating aircraft [2]. All the MPD requirements, together with additional airline requirements, are translated into scheduled tasks that can be executed by airline mechanics. These tasks are listed in the Aircraft Maintenance Manuel (AMM) and organized by the standardized Air Transportation Association (ATA) chapter structure, which breaks down the entire aircraft into different systems followed by related sub-systems, making it simpler for mechanics to find information on particular systems for different aircraft types [19].

Aircraft maintenance tasks are performed to assure a safe, reliable, and airworthy aircraft. When an aircraft enters into operation its designed state naturally deteriorates over time, shown in Figure 2.1a. After a while, the aircraft's current state reaches a certain threshold where maintenance actions are necessary to prevent the aircraft from failing. This type of maintenance is called preventive maintenance, also referred to as scheduled maintenance, and is indicated in Figure 2.1b by points a and b. It can also happen that the aircraft deteriorates faster than planned and exceed the predefined threshold. In worst-case scenarios, the aircraft breaks down and cannot be operated anymore. Maintenance actions are then needed to restore the aircraft to an operational state. These moments are unpredictable and the employed maintenance actions are, therefore, categorized as unscheduled maintenance [2]. This phenomenon is illustrated by points c and d in Figure 2.1b.

Different maintenance concepts, including scheduled and unscheduled maintenance, were identified by Tinga [4] and summarized in a tree structure, illustrated in Figure 2.2, by Oudkerk [3]. From the tree structure, it can be seen that there are three main concepts: Reactive, Proactive, and Aggressive. Reactive and Proactive maintenance is more focused on maintaining the aircraft when it is in operation, while the aggressive direction touches upon the design improvements of the aircraft, which can result in fewer maintenance events during its operational phase. This branch is not of interest for the rest of this literature review as this study focuses on spare parts for current operational aircraft. The Reactive and Proactive concepts are further broken down into multiple sub-concepts, including unscheduled (corrective) maintenance and scheduled (preventive) maintenance, which are discussed in Subsection 2.1.1 and Subsection Subsection 2.1.2, respectively.



(a) The natural deterioration of a system

(b) Difference between scheduled and unscheduled maintenance

Figure 2.1: Deterioration of an aircraft component [2]

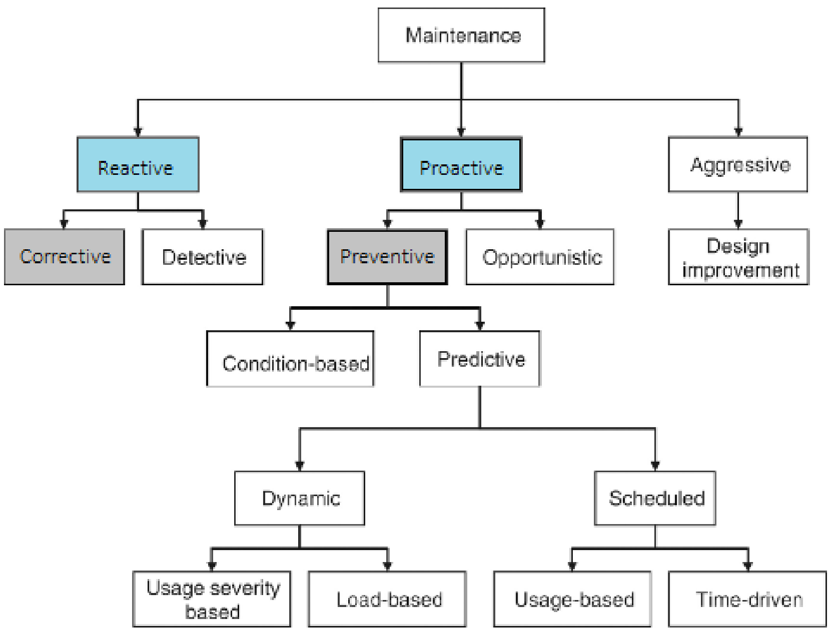


Figure 2.2: Overview of different aircraft maintenance types adapted from Oudkerk [3], based on a study by Tinga [4]

2.1.1. Unscheduled Maintenance

Reactive maintenance, most of the time referred to as unscheduled maintenance, is the unplanned failure of a system or component of an aircraft. This non-routine maintenance is unpredictable and depends on many factors. It can be that for example an unrelated component is damaged during a scheduled maintenance event; Engine components are damaged due to a bird strike; Or a system deteriorates faster than planned. In all these cases, maintenance actions are directly needed to restore the aircraft to an operable state in order to limit its unavailability. Non-routine maintenance cases are also known as Aircraft On Ground (AOG) situations.

The outdated Maintenance Steering Group-2 (MSG-2) approach, a process-orientated method that develops the MRBR, identifies 3 different Primary Maintenance Processes (PMP), which are still being used in older aircraft models and somehow incorporated into the newer MSG-3 approach (task-orientated approach): Hard Time (HT), On-Condition (OC) and Condition Monitoring (CM). The first two processes are related to scheduled maintenance and further discussed in Subsection 2.1.2. For CM components there are no predefined thresholds or requirements that indicate when the component should be replaced to prevent failure. These components are, therefore, only replaced after failure and thus during an unscheduled maintenance event [2]. CM processes monitor operational data of components to analyze the failure rates and their behavior to eventually implement corrective procedures [20].

2.1.2. Preventive Maintenance

Preventive maintenance can either be based on predefined schedules or on opportunistic moments that are beneficial for an aircraft operator. Scheduled maintenance events are all listed in the MPD and prevent the aircraft from failing during its operation. For most aircraft types the maintenance events are grouped into four different letter checks, based on the utilization of the aircraft. This utilization is based on the number of calendar days (CD), flight cycles (FC), and/or flight hours (FH). An overview of the scheduled letter checks is given below in Table 2.1. Newer aircraft models that follow a modified MSG-3 approach have a more dynamic interpretation of scheduling tasks compared to the traditional letter checks. In this approach, the maximum FH, FC, or CD intervals are independently identified for every system, which creates a more adaptable schedule. However, some operators still choose to combine these tasks into (letter) blocks to simplify their scheduling problem [2].

Table 2.1: Overview of standard scheduled maintenance checks [7, 16, 17]

Check	Interval	Description
Daily (transit)	After each FC with a turnaround time of more than 4 hours	A basic visual inspection of the aircraft on deterioration or damages.
A-check	2-4 months	General visual inspection of the aircraft, including some servicing.
B-check	-	Detailed check on the aircraft components and systems.
C-check	16-24 months	An extensive maintenance check that includes a functional operational check on different systems and components. The aircraft must be overhauled for several days. Special equipment and trained mechanics are necessary to perform the check. C-checks automatically include A and B checks.
D-check	6-12 years	heavy maintenance checks are maintenance checks where the entire aircraft is taken out of service for a longer period. Sections of the aircraft are dissembled for structural inspection of components on corrosion, cracking, deterioration, and all kinds of structural damage. A, B, and C-check are also performed during this check.

As already mentioned in Subsection 2.1.1, there are 2 PMP that were identified by the MSG-2 approach and fall under scheduled maintenance activities, namely HT and OC:

- **Hard Time (HT):** This process belongs to the preventive maintenance branch and is applicable to components/systems with predefined life limits. These life limits are expressed in the form of FH, FC or CD [26]. The HT process requires the removal and replacement of components before the predefined life limit exceeds. Sometimes it is possible to restore the removed components. In general, the HT process includes components/systems that are critical within the aircraft [2].
- **On-Condition (OC):** Similar to the HT process, the OC process also belongs to the preventive maintenance category. However, this process includes components/systems where wear-out/deterioration is detectable. In most cases, inspections are necessary to detect if the current state of the component or system is still within the predefined interval and not below or above a certain threshold, depending on the component/system [20]. The wear-out inspections take place after a predefined number of FH,

FC, or CD, and the results of the inspections are used to determine the remaining serviceability of the component/system [2].

2.2. Spare Parts Characteristics

During maintenance activities, aircraft parts are either replaced with new, repaired, or serviced parts. These spare parts have different characteristics, which are classified via different approaches. Traditionally, aircraft spare parts are categorized into four types based on a combination of maintenance regulations, lifespan, economic value, and functional criteria. These types are known as Rotables, Repairables, Expendables, and Consumables. Each type is described below [19, 27, 28]:

1. **Rotables:** These are the most expensive and complex items of an aircraft. The parts have a unique assembly serial number and are serviced and repaired during a dedicated maintenance event. It is expected, under normal conditions, that the part never has to be fully replaced with a new part and its functionality can be guaranteed by unlimited repairs. The landing gear and major engine components are examples of rotable items.
2. **Repairables:** These items can be repaired for a limited number of times after which they will be replaced with a new part. The decision to repair or replace an item depends on the technical/structural condition with respect to the economic benefits. Examples of repairable items are tires, passenger seats, and engine blades.
3. **Expendables:** Items that are always replaced after removal. These parts are usually very standard and routinely consumed during maintenance activities. Repairing these parts is not economically beneficial due to their relatively low cost. Examples are cotter pins, screws, and rivets.
4. **Consumables:** Items, normally raw materials/chemicals like oils, wires, lubrication, etc., that are consumed during aircraft operations.

Next to the main general differences in part type, another distinction can be made concerning the criticality of the part. Not all items directly influence the operational functionalities of the aircraft. Consequently, it may not always be necessary to suspend the aircraft from commercial operations and perform immediate maintenance activities to restore a specific part. In some cases, a failure of a specific item may still be within the safety margins for a permitted flight. However, a combination of failed parts could lead to an AOG situation, necessitating immediate action.

For every aircraft type, a Master Minimum Equipment List (MMEL) is available, which defines the operational status of the aircraft prior to take-off [18]. Similarly, spare parts are assigned an Essentiality code (ESS) to indicate whether a defective part needs immediate replacement or if it does not directly impact the operation and safety of the aircraft [19]. The ESS codes are described below in Table 2.2.

Table 2.2: Spare parts essentiality codes and their impact on the aircraft operational status [18, 19]

ESS code	ESS	Description
1	No-Go	Commercial operation of the aircraft is permitted under all conditions, despite the failure or absence of this item.
2	Go-If	Operation of the aircraft is only permitted under specific conditions and for a limited amount of time when this item is defective.
3	Go	The aircraft is operable under all conditions when this item is defective.

Identical or relatively similar parts can be produced by different part manufacturers as long as the manufacturer has a Part Manufacturer Approval (PMA). This approval is granted by the FAA and ensures the same functional requirements as the OEM part. Furthermore, some parts can also be substituted with alternative (non-unique) parts while still fulfilling the same requirements. A part can be one-way interchangeable, part A can replace part B but not vice versa, or two-way interchangeable, part A can replace part B and vice versa. The interchangeability between spare parts is illustrated below in Figure 2.3.

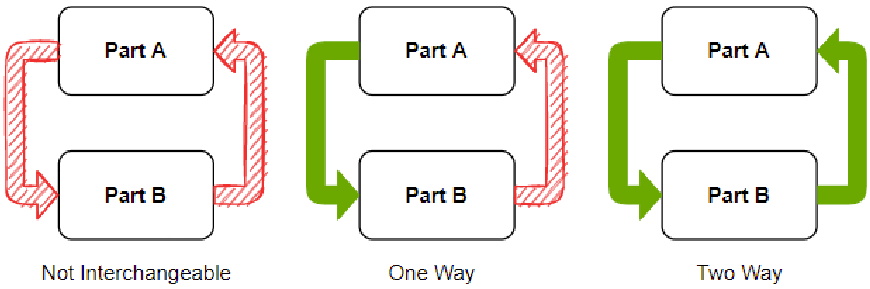


Figure 2.3: Three different interchangeability types for aircraft spare parts [5]

Lastly, spare parts are sometimes listed as proprietary parts. These parts are exclusively manufactured by or for a manufacturer and sold solely by the same manufacturer. Resulting in a monopolistic market for those specific spare parts. The advantages are that all products are traceable, and prediction models can be more precise since they do not have to take other sellers into account, which reduces the unknown variables compared to a competitive market with more uncertainties.

2.3. Overview of the Aircraft Maintenance sector

Various stakeholders are involved within the aircraft maintenance sector, including Aircraft OEMs, such as Airbus or Boeing, Part OEMs, aftermarket distributors, and Maintenance Repair and Overhaul (MRO) services. The diagram below in Figure 2.4 provides a schematic overview of the interactions between these stakeholders.

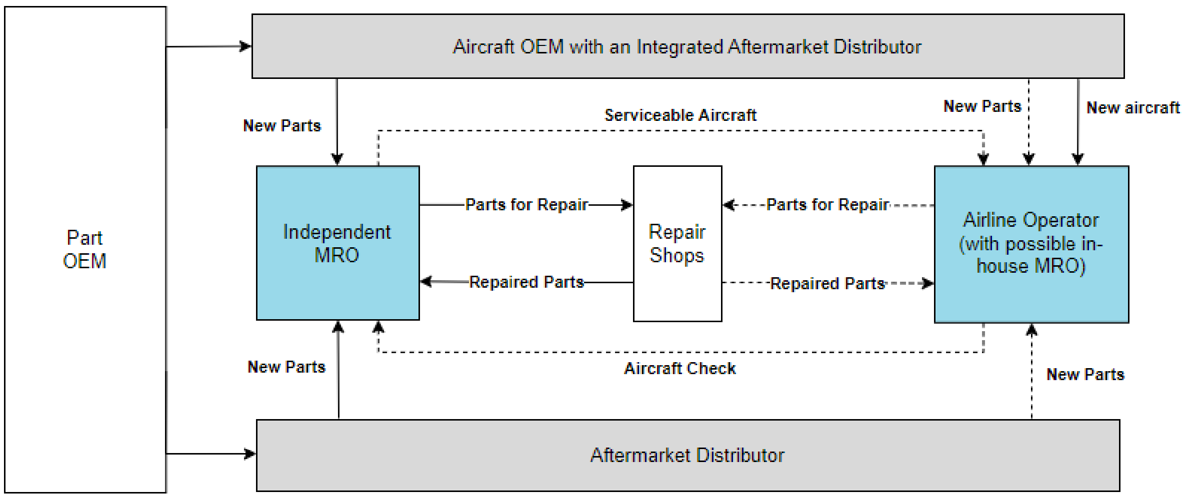


Figure 2.4: Simplistic overview of the interactions within the aircraft maintenance industry, adapted from Jan Gottemeier [6] and based on a study by Rodrigues Vieira and Loures [7]

Part suppliers, represented by the term Part OEM in Figure 2.4, manufacture and supply spare parts to aftermarket distributors and Aircraft OEMs. Aftermarket distributors subsequently distribute these parts to MRO organizations, which can be independent entities or airline operators with in-house MRO services. The interaction between airline operators and independent MRO services depends on the chosen MRO strategy of the airliners. Aircraft OEMs, on the other hand, utilize the obtained spare parts from the part supplier to assemble new aircraft. A broader description of the MRO services and aftermarket sector is given below in Subsection 2.3.1 and Subsection 2.3.2, respectively.

2.3.1. Maintenance Repair and Overhaul (MRO) Sector

The term MRO includes all related actions to maintain an aircraft to guarantee the intended functions of all items (components, systems, etc.) necessary to operate in a safe and reliable environment [2]. MRO organizations are responsible for carrying out these maintenance activities. Airliners generally adopt one of the four different MRO strategies, which differ in terms of inventory control and maintenance tasks: Fully Integrated MRO, Partially Outsourced MRO, Mostly Outsourced MRO, and Wholly Outsourced MRO. A visual representation of these strategies is provided in Figure 2.5.

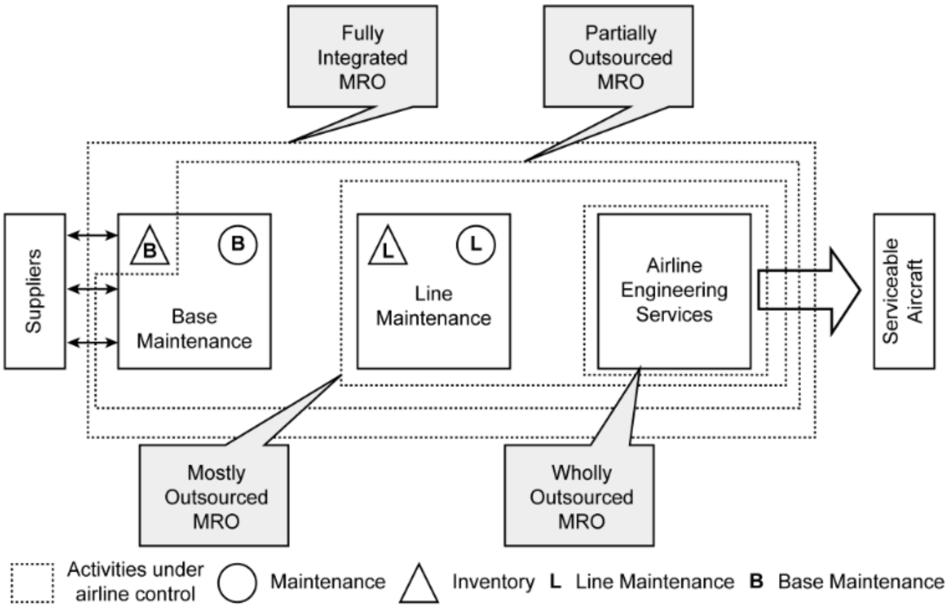


Figure 2.5: A schematics overview of different MRO strategies for airline operators, adopted from Al-Kaabi et al. [8].

Airliners that adopt, for example, a Fully Integrated MRO strategy are executing both line and base maintenance tasks, as well as controlling their corresponding spare parts inventory levels. Whereas other airliners choose to partially or even completely outsource all possible MRO activities [8]. Line maintenance tasks include all maintenance activities that can be performed during the aircraft's turnaround time, while more complex and time-consuming hanger/base maintenance tasks are only performed on aircraft that are out-of-service [2].

An airliner's decision on which MRO strategy to adopt is mainly based on economic reasons. For airliners with a large fleet, consisting of different aircraft types and various operating destinations, it is beneficial to keep all MRO activities in-house (Fully Integrated MRO). In many cases, these airliners developed their MRO business into an independent subsidiary which enables them to offer services to other airliners as well, resulting in less expenses. On the other hand, new airliners are more inclined to outsource their entire MRO activities as it is more cost-efficient. Other reasons why airliners can choose to partially or fully outsource their MRO activities are listed below [8]:

- **Specialized skills:** Airline operators may have limited access to knowledge or employers with special expertise, whereas MRO organizations, with their primary focus on maintenance, typically possess such expertise in-house. Similarly, MRO organizations are better equipped with specialized tools that are required for certain maintenance activities.
- **Repair time:** MRO organizations are more experienced in conducting different maintenance tasks and can handle multiple maintenance events simultaneously, leading to quicker task completion.
- **Focus shifting:** Airliners can focus more on other non-core businesses when the MRO is outsourced. Furthermore, low-cost carriers only operate new aircraft and are constantly updating their operational fleet. Therefore, it is unnecessary to invest in specialized equipment and infrastructure to conduct hangar maintenance events.

2.3.2. Aftermarket sector

MRO organizations typically procure necessary spare parts from aftermarket companies, which serve as distributors for a wide range of aircraft spare parts. The ordering process at an aftermarket distributor should be taken into account by the MRO organization, as it can be time-consuming due to potential product unavailability or the need for spare part certification procedures. A simplistic overview of an order process at an aftermarket is illustrated below in Figure 2.6. The gray box in the top-right corner of the flowchart represents the additional steps required to fulfill an order when a product is currently unavailable/out of stock at the aftermarket distributor.

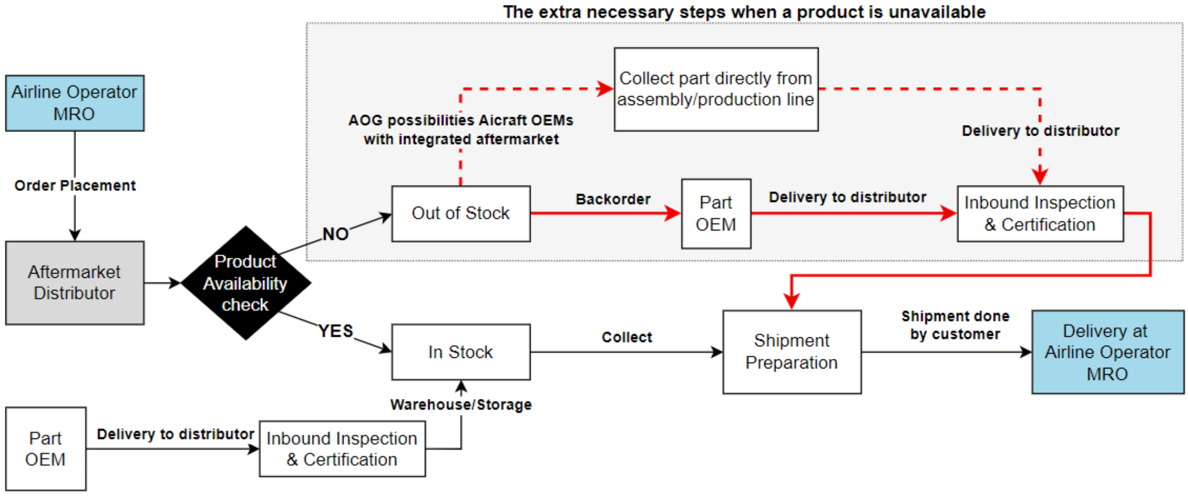


Figure 2.6: Complexity of an order process at an aftermarket company

If a requested product is out of stock, the aftermarket distributor must backorder the requested product at a Part OEM. The manufacturer will then, if the requested product is available or in production, ship the product to one of the aftermarket warehouses, where an inbound inspection will take place to ensure its quality and functional requirements. In the worst-case scenario, when the product is also unavailable at the Part OEM, the manufacturer first needs to set up an entire production line which may increase the delivery time by several months.

In situations where the aftermarket company is a subsidiary of an aircraft OEM, it is sometimes possible to improve the delivery time by directly taking out the requested part from the assembly line of an aircraft. However, it is important to note that this is far from optimal and is only implemented when necessary, depending on the agreements between the airline operator and the aircraft OEM/aftermarket. This order process is represented by the red-dotted line in Figure 2.6

2.4. Conclusion

In general, aircraft maintenance can be classified into scheduled and unscheduled events. Scheduled events are planned by an airliner or MRO, and involve various checks of the aircraft based on its utilization, which is usually expressed in Flight Hours (FH), Flight Cycles(FC), or Calendar Days (CD). While scheduled maintenance events are performed to prevent an aircraft from failing, unscheduled events are unpredictable and usually executed directly. During these non-routine maintenance events, also known as Aircraft On Ground (AOG) situations, a component is replaced to restore the aircraft to a safe and airworthiness condition.

Modern aircraft are composed of thousands of parts that all have to be maintained regularly to guarantee a safe and reliable aircraft. These parts vary in many characteristics and are categorized via different approaches. Traditionally, all parts are categorized based on their value, lifespan, and functional criteria, into 4 types: Rotables, Repairables, Expendables, and Consumables. The importance of a part is indicated by an essentiality code, which describes whether an aircraft is permitted to take off when that part is defective or missing. Furthermore, parts can also be interchangeable with each other.

Various parties are involved in aircraft maintenance. Airliners adopt different strategies with respect to outsourcing their maintenance activities to MRO organizations. The aftermarket sector serves as a distributor of spare parts to these customers, but the ordering process can be complex due to the diversity and availability of spare parts. Overall, when considering the different characteristics of aircraft maintenance and spare parts, as well as the involvement of different entities, executing a maintenance task is often more complex than expected.

Predicting the need for spare parts during a selected maintenance event is difficult for most maintenance organizations, especially for smaller airliners with limited resources. However, Aftermarket distributors that are integrated with an Aircraft OEM can combine the technical knowledge from the maintenance documentation with previous sales records of spare parts to generate an overview of necessary spare parts. These combined components form the foundation of this MSc Thesis project. This research will address the complexity of aircraft spare parts demand predictions and aims to develop a robust prediction model for an aftermarket e-commerce service that estimates the need for correlated spare parts to generate a complete bill of material based on previously ordered parts.

3

Spare Parts Demand Forecasting Models

One of the main challenges in the aviation sector is accurately forecasting spare part demand for maintenance activities. Various forecasting techniques have been proposed to increase accuracy, all with their own unique characteristics. This chapter focuses on demand forecasting models for spare parts. First, classification schemes to identify different demand patterns, which influence the performances of forecasting models, are investigated in Section 3.1. Followed by a comprehensive overview and description of parametric forecasting techniques in Section 3.2. After which, popular non-parametric models are discussed in Section 3.3. This chapter ends with a short recap in Section 3.4.

3.1. Demand Patterns

Analyzing and classifying demand patterns for spare parts is useful as it helps in understanding specific characteristics that can improve forecasting accuracy and stock control by choosing a better prediction model [10]. Classification schemes group multiple spare parts with similar characteristics together. Next to the forecasting enhancements, it enables managers or companies to shift their attention to the most important classes [29]. Traditionally, classification techniques can be divided into mono-dimensional categorization schemes, where classification is done on a single variable, and multi-dimensional categorization schemes, where multiple variables are used to classify the demand pattern. However, it is also possible to make another distinction by focusing more on quantitative or qualitative techniques [30].

Qualitative approaches are based on the judgment of experts and aim to classify parts into different categories concerning their usage, storage constraints, costs, essentiality code, and operational influences such as an aircraft's downtime due to part replacement. A commonly used technique is the VED method, which stands for Vital, Essential, and Desirable. Experts label the most crucial parts as vital, whereas the less important parts are labeled as desirable. An Analytical Hierarchic Procedure (AHP) helps with the assignment procedure, as it may suffer from the subjective decisions of experts [30]. Quantitative classification approaches are more of interest for this literature review as it focuses on analyzing numerical data patterns. Therefore, Qualitative approaches will not further be reviewed in this study.

Quantitative Classification Techniques

Quantitative techniques are popular used techniques for spare part classification as they focus on analyzing numerical data patterns. One of the earliest developed classification schemes is the ABC classification model, which groups spare parts into three different classes based on the total annual sales volume (cost per unit times the annual sold units) [31]. Apart from mono-dimensional ABC schemes, several studies suggest that adding extra variables, such as part criticality [32] or the number of customer transactions [33], could improve the ABC classification.

Over the years, multiple different classification techniques have been proposed. Williams [34] was one of the first who specifically considered variance participation. A technique that classifies spare parts based on the analysis of historical demand variation during lead time into smooth, slow-moving and sporadic classes. Each category was associated with the most appropriate forecasting model, resulting in lower inventory costs compared to a generic model that uses a continuous distribution for all parts. Syntetos et al. [9] further developed the Williams' scheme by introducing a two-dimensional matrix to classify spare parts into four main categories based on the average demand interval (ADI) and squared correlation of variation (CV^2) [30]. This classification scheme is widely used in the aviation industry and commonly known as the Syntetos-Boylan-Croston scheme. A schematic representation can be found in Figure 3.1, followed by the definition of the four categorization classes: Smooth, Erratic, Intermittent, and Lumpy.

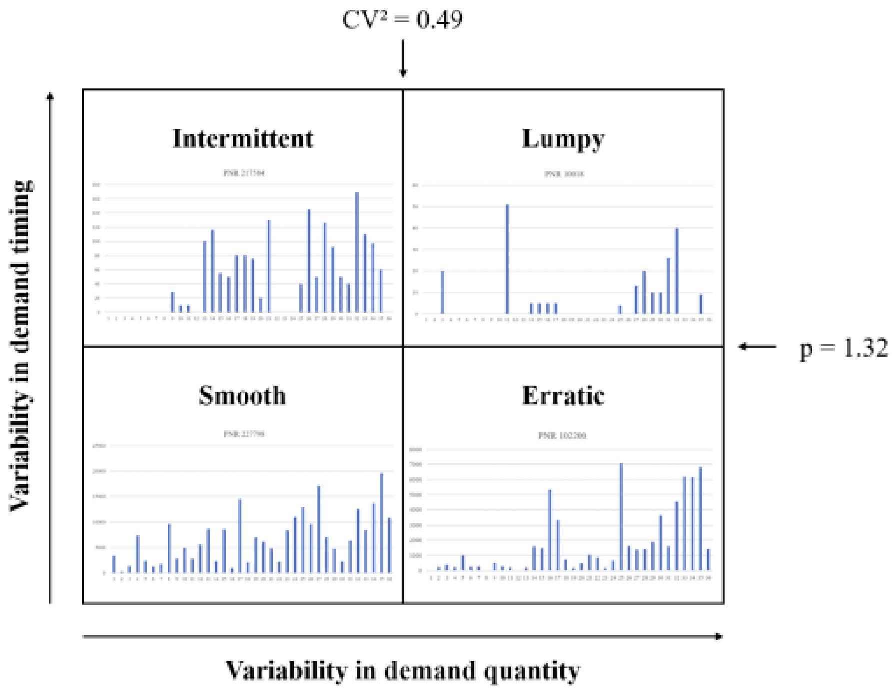


Figure 3.1: Spare part classification technique and the corresponding cut-off values based on a study from Syntetos et al. [9]

- **Smooth:** Constant demand interval and almost no variation in demand quantity. Usually, these patterns are easier to forecast. There is no risk of overstocking as the items are sold regularly.
- **Erratic:** Demand distribution over time is constant and comparable to the smooth pattern. However, there is a large variation in demand size.
- **Intermittent:** Many periods with zero (no) demand occurrences. The demand size is constant over time.
- **Lumpy:** Extremely sporadic demand, many periods with no demand at all. High variation in demand size.

A crucial aspect of the classification process is the determination of the cut-off values for each class. Previous studies determined these cut-off values arbitrarily or subjectively based on judgmental decisions, after which different forecasting techniques were tested on all categories to identify the most suitable prediction model for each category. Syntetos et al. [9] approached this problem differently by comparing the theoretical quantified error measures of three different parametric forecasting techniques to understand their underlying performance characteristics. These forecasting techniques, namely Croston, Single Exponential Smoothing (SES) and Syntetos-Boylan-Approximation (SBA), are explained in Section 3.2. Their analysis revealed that the optimal cut-off values for ADI and CV^2 were found to be 1.32 and 0.49, respectively. To validate their findings, they conducted demand forecasts for 3000 different automotive spare parts, and the results demonstrated that the classification scheme could indeed be utilized to select the most suitable forecasting method.

The cut-off values determined by Syntetos et al. [9] are commonly used to categorize spare parts as smooth, intermittent, erratic or lumpy [35]. Due to the uncertainty in demand size and interval, most aircraft spare parts encounter a lumpy or intermittent demand pattern, which is challenging for accurately predicting demand [20].

3.2. Parametric Time series methods

Parametric forecasting models fit given data points to a known probability distribution and utilize the characteristics of the fitted distribution to predict future data points [10]. One of the earliest used parametric forecasting techniques is the Simple Exponential Smoothing (SES) method, developed by Brown [36] in 1956. This univariate technique is commonly used to forecast demand, and is the foundation for later developed models. The original form is considered simple, as it is unable to detect any trends or seasonality. The underlying concept of the SES model is to predict future values by considering the weighted sum of previous observations in chronological order, giving more weight to recent observations compared to older ones. A representation of the model is shown in Equation 3.1. The outcome of the model, S_t, stands for the predicted value at timestamp t which is calculated by multiplying the smoothing constant with the currently observed value, which is denoted as X_t. The smoothing constant α, usually a value between 0 and 1, indicates the effect of the current observation on the predicted value. The closer α is to 1, the higher the effect of the previous observation. When α is smaller more previous observations are considered to determine future values [6, 37].

$$S_t = \alpha X_t + (1-\alpha)S_{t-1} \tag{3.1}$$

Holt [38] developed, independently from Brown and around the same period, a Double Exponential Smoothing (DES) model. This model is similar to the SES model but also accounts for trend effects in the prediction. The expression of the model is given Equation 3.2. A constant trend smoothing parameter, expressed by γ, is used to generate the smoothed additive trend at the end of period t, which is noted by T_{t-1}. This additional term is added to the original SES to include the trend effects in the predictions[37].

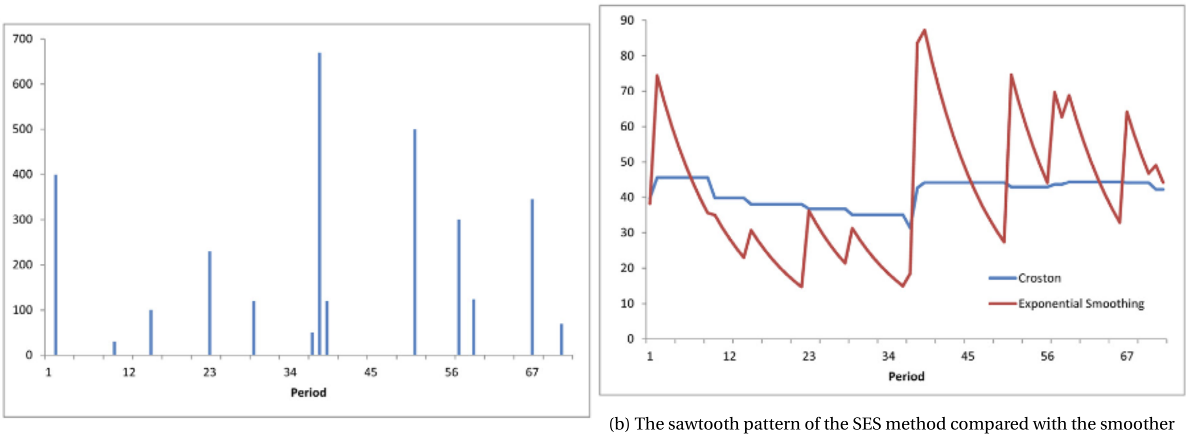
$$\begin{aligned} S_t &= \alpha X_t + (1-\alpha)(S_{t-1} + T_{t-1}) \\ T_t &= \gamma(S_t - S_{t-1}) + (1-\gamma)T_{t-1} \end{aligned} \tag{3.2}$$

Holt [38] also developed a third forecasting model for time-series that are suffering from seasonal patterns. This model is known as the Triple Exponential Smoothing (TES) model and is formulated in Equation 3.3. Two additional variables are included to capture seasonality: The smoothed seasonality index, I_t, calculated at the end of time period t; And the constant smoothing seasonality index (δ). Winters tested both Holt's models to prove their theoretical concepts. Nowadays, these models are known as the Holt-Winter (HW) Models. It is also possible to combine the expressions for trend and seasonality into one model [37].

$$\begin{aligned} S_t &= \alpha(X_t - I_{t-p}) + (1-\alpha)S_{t-1} \\ I_t &= \delta(X_t - S_t) + (1-\delta)I_{t-p} \end{aligned} \tag{3.3}$$

3.2.1. Croston's Method and its Modifications

Croston discovered that the SES methods are biased when demand data follows an intermittent demand pattern, resulting in inaccurate estimates leading to overestimated stock levels. The problem of the SES model is that it only generates new values after a period with nonzero demand [10]. Most recent demand periods are affecting the predicted value the most, resulting in higher predicted demand values just after a demand occurrence. These overestimated demand values are decreasing during zero demand periods [39]. This SES bias is well illustrated by Pinçe et al. [10] in their critical review on intermittent demand forecasting. The authors described the SES bias as sawtooth pattern, which is illustrated by the red line in Figure 3.2b. The predicted demand values from Figure 3.2b are the result of the observed demand values from Figure 3.2a.



(a) The real demand observations over a time-period

(b) The sawtooth pattern of the SES method compared with the smoother obtained results from the Croston method.

Figure 3.2: True values vs predicted values by the SES and Croston method, adopted from Pinçe et al. [10]

To solve the bias of the SES, Croston [40] developed a new method that could deal with intermittent demand patterns by decomposing the demand prediction into demand sizes and inter-demand periods. Both components are separately modeled with the SES method and later combined to predict the demand value per period [10]. This method is known as Croston's Method and became an important benchmark for other studies on spare part demand forecasting. Both components, the inter-demand period and demand size indicated by $\hat{p}_t$ and $\hat{z}_t$, respectively in Equation 3.4, are only updated after a positive demand occurrence ($z_t \neq 0$). If no demand occurs ($z_t = 0$), the predicted values are constant and identical to the generated demand size in the previous time-period. Eventually, the demand forecast itself can be calculated by dividing z_t over p_t, as expressed in Equation 3.5 [10, 41].

$$\begin{aligned} z_t = 0: \quad & \hat{p}_t = \hat{p}_{t-1}, & \hat{z}_t = \hat{z}_{t-1} \\ z_t \neq 0: \quad & \hat{p}_t = \alpha p_t + (1-\alpha)\hat{p}_{t-1}, & \hat{z}_t = \alpha z_t + (1-\alpha)\hat{z}_{t-1} \end{aligned} \tag{3.4}$$

$$\hat{Y}_t = \frac{\hat{z}_t}{\hat{p}_t} \tag{3.5}$$

Pinçe et al. [10] used the same demand observations, illustrated in Figure 3.2a, to compare the SES model with Croston's method. Figure 3.2 shows that with Croston's method, the sawtooth pattern disappears and the demand forecast becomes smoother. Furthermore, when comparing Equation 3.4 with Equation 3.1, it can be concluded that Croston's method is identical to the SES model if there are demand occurrences in every period [41].

Multiple studies tested both methods and compared their accuracy. Willemain et al. [42] concluded that the Croston method is superior to the SES method under intermittent demand conditions. The same conclusion was drawn by Eaves and Kingsman [43], after testing and comparing different forecasting models on predicting aircraft spare part demand for the UK Royal Air Force. The Croston method is one of the most popular forecasting methods used in the aviation industry for spare part demand prediction and can be found in multiple leading software packages [41].

Modification on Croston's Method

Syntetos and Boylan [41] showed that Croston's method is still slightly biased due to the assumption that the estimators of both components, $\hat{z}_t$ and $\hat{p}_t$, are independent. The error's magnitude is directly related to the value of α, which is used to calculate $\hat{p}_t$. To overcome this error, Syntetos and Boylan [41] proposed a modified version of Croston's method by adding a bias correction coefficient, shown between brackets in Equation 3.6. This modified version of Croston is known as the Syntetos-Boylan-Approximation (SBA) in literature. In order to evaluate the SBA performance, Syntetos and Boylan [41] tested their solution on a dataset containing 3000 different spare parts for the automotive industry. Their analysis showed its superiority over SES and Croston for fast intermittent demand patterns [10].

$$\hat{Y}_t = \left(1 - \frac{\alpha}{2}\right) \frac{\hat{z}_t}{\hat{p}_t} \tag{3.6}$$

Another modified Croston method is the Tuenter-Syntetos-Babai (TSB) model, developed by Teunter et al. [44]. This method specifically focuses on spare parts that are affected by demand obsolescence. The problem with the original Croston method is that it is only updated after a demand occurrence, resulting in outdated demand values for periods with zero demand, making it impossible to detect or approximate demand obsolescence. Teunter et al. [44] solved this problem by updating the demand values periodically. In addition, The TSB model combines the demand size forecasts with the demand probability forecasts instead of the inter-demand forecasts to predict upcoming demand occurrences. This change should improve the reaction time of the model if demand obsolescence originates.

Hemeimat et al. [45] compared the performances of the TSB model with Croston, SES, and SBA in forecasting monthly orders for 1200 spare parts from a paper mill company. The spare parts were classified as non-moving, slow-moving, or fast-moving items. The analysis showed that the TSB model achieved lower Mean Errors (ME) and Root Mean Square Errors (RMSE) for slow and non-moving parts. However, for predicting the upcoming orders of fast-moving items, the SBA method achieved lower errors.

3.2.2. Alternative Parametric Models

Although Croston's method and its respective variants are the most commonly used methods within the industry, some other methods are interesting to consider for intermittent or lumpy demand forecasting. One such method is the ARMA model, which was developed by Box et al. [46]. The model consists of two polynomials: An Auto Regressive (AR) polynomial to capture the dependencies between the observations; And a Moving Average (MA) model to linearly forecast the relation between demand observations and previous residual error terms. Multiple studies showed that the results are comparable to SES and HW models [47–49].

A more promising forecast method for intermittent demand patterns is the newer ARIMA model, a modified version of the ARMA model that includes an integration component to remove the non-stationarity within the time-series. Gautam and Singh [48] demonstrates that the ARIMA model outperforms the ARMA model in predicting monthly air passengers in terms of the Mean Absolute Percentage Error (MAPE). However, the same study concluded that the ARMA model performs slightly better when predicting hourly energy consumption. Both models have comparable performances to the SES model.

Another alternative approach for forecasting spare parts demand is the recently developed Prophet method [50]. This method consists of three different components to capture trends, seasonality, and event effects. Unfortunately, research on the model performances with respect to intermittent and lumpy demand data patterns is still limited. Nevertheless, Jan Gottemeier [6] concluded that the prophet method is comparable to other traditional parametric methods when forecasting demand for aircraft spare parts that exhibit lumpy patterns. The same conclusion was drawn by Sousa et al. [51] when they tested the methods on predicting customers arrivals at stores with an hourly frequency between January 2015 and November 2019.

3.3. Non-parametric Models

Parametric models are considered to be simple and straightforward to use. However, there are multiple disadvantages when it comes to these models. First, parametric models can easily misinterpret the relation between dependent and independent variables. Resulting in modification inflexibility during their design phase [52]. Second, outliers can affect the model, which leads to biased parameters. Recalibration of the model is needed to overcome this problem [53]. Lastly, parametric models have difficulties in recognizing non-linearity, which often occurs in intermittent or lumpy demand patterns [54]. Non-parametric models overcome these drawbacks as these models do not dependent on a probability distribution but try to reconstruct the empirical distribution, making them more flexible and better suited for forecasting intermittent and lumpy demand patterns [10]. This section mainly focuses on Artificial Neural Networks (ANNs) and their applications.

3.3.1. Artificial Neural Networks

Artificial Neural Networks (ANNs or NNs) are computational models that are inspired by the human/animal neuron system. Most conventional computing systems are performing calculations with higher computational and can outperform human brain capacity. The problem with these conventional models is that they are only applicable to a specific given problem, while ANNs structures can solve different complex problems with simple computational operations by also looking into the organization of elements and their correlations. The human nervous system works relatively similar. During the learning phase, the connections between neurons are continuously adapting into denser, thinner, newly formed, or disappearing connections depending on the relations between neurons. A simplified schematic comparison of a biological neuron with an ANN is illustrated in Figure 3.3. The input (X) can be seen as the dendrites of biological neurons, which are triggering the activation of the neurons. The dependencies between the neurons are indicated with different weights (W) and represent the axon of the human nervous system. The model's output is comparable to the Synapse and is computed by the cell body, which can be seen as the computational power of the neuron. These different weighted connections between neurons form the fundamental principles of ANNs [55].

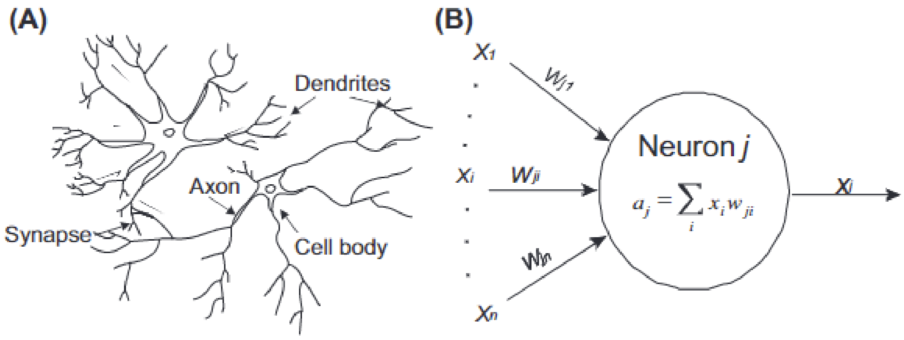


Figure 3.3: MLP network used by Park and Lek [11]

NNs are not based on distribution assumptions and the relations between neurons are easily adaptable when new information is introduced, which makes them extremely flexible in modeling linear and non-linear patterns [54, 56]. According to Wieland and Leighton [57], These two aspects are the most powerful aspects of ANNs. However, it is important to notice that ANNs could get stuck in a local optimum, which affects the final outcome [58]. The biggest challenge when using ANNs is related to the availability of historical data, which is necessary to train the model in order to obtain desired accuracy levels [30]. Unfortunately, the amount of training data can be limited, especially when many stakeholders are involved, which is the case for the aviation sector.

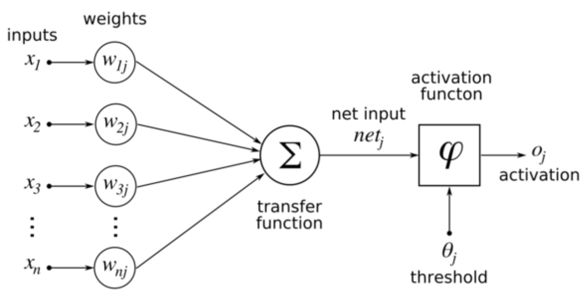
Over the years, several types of ANNs have been developed, each with their own specific capabilities to obtain the best results depending on the problem characteristics. In general, when considering the learning phase of the models, a distinction between supervised and unsupervised learning models can be made. Supervised learning is applied when the input and output of the model are classified. The ANN is then trained to predict the output based on the performance feedback it receives from the classified output. Unsupervised learning is applied when the output is not classified. These models categorize the dataset by analyzing all the respective data properties and are commonly used to cluster unlabeled datasets [11]. Supervised learning ANNs are more common for time series forecasting. Examples of frequently used methods for demand predictions are discussed below, together with their capabilities and design flaws.

Multi Layer Perceptron (MLP) Model

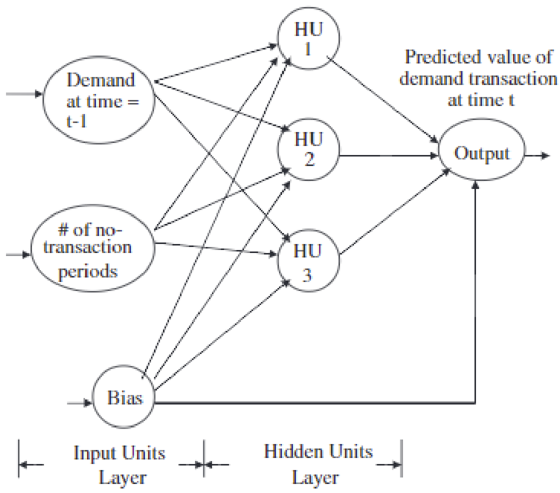
The Perceptron, proposed by Rosenblatt [59], is one the first developed artificial neurons and has become a fundamental element in many ANNs. A simplistic diagram of a Perceptron is presented in Figure 4.2a. The inputs are multiplied with their corresponding weights and summed by the transfer function, and represent the net input of the neuron(net_j). The activation function transforms the net_j into the output of the neuron (o_j) to capture, process, and recognize the complexity of the data, to enhance the learning capabilities of the network. Various activation functions can be used to constrain the output within certain limits, such as a linear or threshold activation that can model linear patterns. A more advanced activation function is the Sigmoid function. This function is popularly used by programmers as it captures non-linearity. Other non-linear activation functions are Softmax, Tangens hyperbolicus (Tanh), Swish, Rectified Linear Unit (ReLU), and different variances of the ReLU activation function. Deciding which activation function to use is complicated as it depends on the context of the problem [55, 60].

For demand forecasting tasks, it has been observed that the ReLU activation function and its variations yield the highest accuracy. In a study conducted by Jan Gottemeier [6], the performance of the same network structure was compared using various activation functions to forecast monthly demand values for aircraft spare parts. The findings concluded that the ReLU function produced the best results. Similarly, Henkelmann [61] analyzed different activation functions for predicting monthly demand values of spare parts manufactured by an automotive OEM and reached the same conclusion in favor of the ReLU function.

A Multi-Layer Perceptron (MLP) network trained by a Back Propagation (BP) algorithm, also known as a multi-layer feedforward neural network, is one of the most commonly used types of ANNs [11]. This model, as well as most ANNs, is built up from multiple Perceptrons. The outputs of the previously layered Perceptrons are the inputs of the next layered Perceptrons. The BP learning algorithm, proposed by Rumelhart et al. [62], minimizes the loss function by adjusting the weights of each layer backwardly. The gradient of the loss function is computed via the chain rule. This process is iterated layer by layer in reverse order, starting from the final layer all the way up to the first layer, which should avoid unnecessary calculations of intermediate parts in the chain rule [63]. The BP learning algorithm was a significant improvement for multi-layer networks as it solved the inefficiencies of the first developed 'multi-layer' training algorithm, which was too slow and strict as it directly calculated the gradient for each weight individually [55].



(a) Overview of the first Neuron proposed by Rosenblatt [59], adopted from [64]



(b) MLP network used by Gutierrez et al. [54]

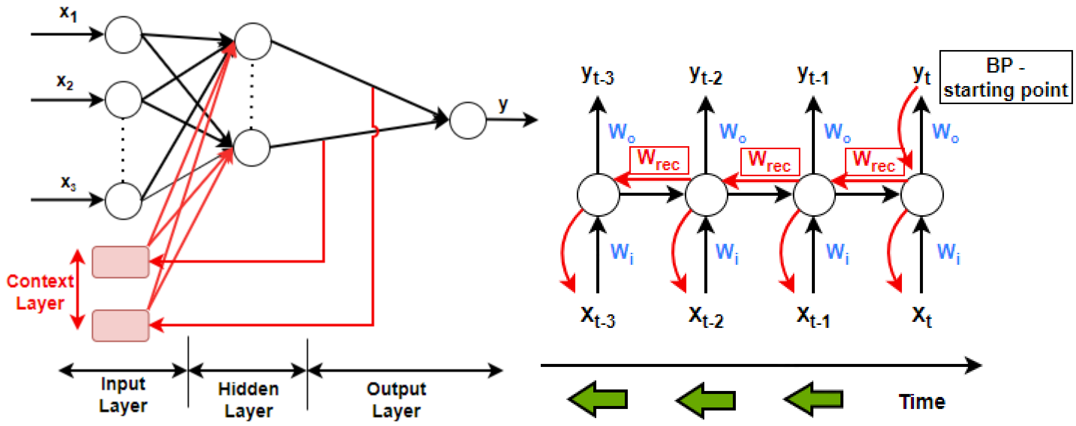
Figure 3.4: Explanation of a MLP network

The simplest form of an MLP consists of three layers: An input layer, 1 hidden layer, and an output layer. This form was used in a study by Gutierrez et al. [54] to forecast daily demand for spare parts of an electronic distributor with a lumpy demand pattern. A schematic overview of their proposed model is illustrated in Figure 4.2b. The input layer consists of two input nodes, each representing an input variable: The demand at the end of the immediately preceding period; And the length, counting from the last nonzero demand period, between the previous two nonzero demand transactions. The hidden layer is constructed out of three Perceptrons and the output layer consists of one output node which equals the predicted demand value.

Gutierrez et al. [54] tested their proposed MLP network on 24 different lumpy demand datasets and compared the obtained results with three parametric approaches: SES, Croston, and SBA. They concluded that their MLP almost always outperforms the parametric approaches, except when there is a significant decrease in the average nonzero demand periods between training and testing data. In this scenario, the parametric models tend to perform slightly better compared to the proposed MLP [30]. This same conclusion was drawn by Mukhopadhyay et al. [39] on the same electronic product demand dataset. Babai et al. [65] extended the input variables with N additional variables, corresponding to the actual demand values at period T minus N. The empirical investigation, conducted on more than 5000 spare parts for an aircraft fleet of an airliner, showed that the extended models are again superior to the parametric models and even outperform the model from Gutierrez et al. [54].

Recurrent Neural Network (RNN)

Rumelhart et al. [62] mentioned, in the same study where they highlighted the MLP network, a new ANN concept that incorporates observations from previous time periods, namely the Recurrent Neural Network (RNN). The RNN's structure is relatively similar to that of an MLP, as can be seen in Figure 3.5a. The main difference lies in the additional context layer that acts as a recurrent loop between the output of the hidden layer and the input layer, illustrated as the red loop in Figure 3.5a. After each iteration, the output signals are fed back into the input layer. Sometimes it is chosen to only reincorporate the errors of the output layer in the recurrent loop. The advantage of this recurrent loop is that the network can also learn from previous observations instead of only considering the feed-forward network, which only considers the current time period's input. This property is particularly useful for forecasting time series and other problems where past data patterns influence current time periods, which is normally the case for demand predictions [55, 66, 67].



(a) A simplistic overview of a RNN

(b) Sketch of the Vanishing Gradient Problem [68]

Figure 3.5: Overview of a RNN and the Vanishing Gradient Problem

Hochreiter [69] and Bengio et al. [70] discovered that RNNs, unfortunately, suffer from the vanishing gradient problem. This problem occurs during the training phase of the network when the BP learning algorithm is applied and particularly affects time series that are influenced by long-term dependencies [12, 66]. An explanatory illustration of the vanishing gradient problem is given in Figure 3.5b. The BP algorithm, as previously explained, is adjusting the weights between neurons by computing the gradient of the loss function. According to the chain rule, when the BP algorithm adjusts the weight of the recurrent loop, indicated by W_{rec} in Figure 3.5b, it has to multiply the same weight multiple times. If the weight is small, the gradient will slowly approach zero and eventually vanishes, which causes difficulties during the training phase of the network as it is unable to update the weights. Conversely, if the initial value for W_{rec} is larger than 1, the gradient will explode as it tends to go to infinity [68].

Amin-Naseri and Rostami Tabar [71] constructed an RNN, similar to the one in Figure 3.5a, to predict the demand for 30 different spare parts from Dassault Aviation. These spare parts are used in business aircraft and exhibit a lumpy demand pattern. Various input features were employed, including the previous demand interval, ADI, CV^2, number of periods separating the previous two non-zero demand interval, and spare part price. Their results clearly demonstrated that the proposed RNN outperforms Croston, SBA, MLP, and a Generalized Regression Neural Network (GRNN). The same conclusion was supported by Lolli et al. [72] in their forecasting analysis of a 24 weekly intermittent demand dataset of spare parts from an automotive company. Additionally, Islam and Ahmed [73] conducted a case study on predicting electrical load demand for a power company and concluded that an RNN model with a ReLU activation function achieved higher prediction accuracy compared to a MLP. Their model also included additional external features, such as holiday periods and climate statistics.

Alternative proposed RNNs in literature for forecasting demand patterns are a Fully Recurrent Network (FRNN) and a Time Delay Neural Network (TDNN). Every layer in a FRNN has a recurrent loop to the previous layer instead of one loop from the the output of the model to the input layer [55]. Wang et al. [74] effectively applied this algorithm to predict building occupancy, which could potentially lead to better facility control and energy efficiency improvements. Their model achieved accuracy levels up to 90% when proper error tolerances were allowed. The TDNN model, on the other hand, is designed to deal with sequential data patterns by extending the input variables with previous periods demand values [72]. Sahin et al. [56] obtained the same results for forecasting the demand over the last 12 months for 90 different aircraft spare parts when using a TDNN instead of a RNN. However, when dealing with erratic demand patterns, The TDNN seems to have a slight advantage over RNN and MLP models in terms of the Geomteric Mean of Mean Absolute Deviation Average (GMAMAD/A).

Long- Short Term Memory (LSTM) Network

Hochreiter and Schmidhuber [75] found a solution to overcome the exploding vanishing gradient problem by proposing a new structure of the RNN and named it the Short Long Term Memory (LSTM) Network. The LSTM adds a Constant Error Carousel (CEC) to each neuron within the network that maintains the error signal in a so-called memory cell. At first, The memory cell consisted of an input and output gate that protects the relevant content from and for a neuron. Later, a forget gate was added by Gers et al. [76], which enables the cell to reset. This addition formed the basis of the LSTM structures that are used nowadays and is known as the Vanilla LSTM. A straightforward overview of the Vanilla LSTM is given below in Figure 3.6, together with some extra explanation on the three included gates. After following all the multiplication and summations steps, one can calculate the new/current cell's value and LSTM's output [12].

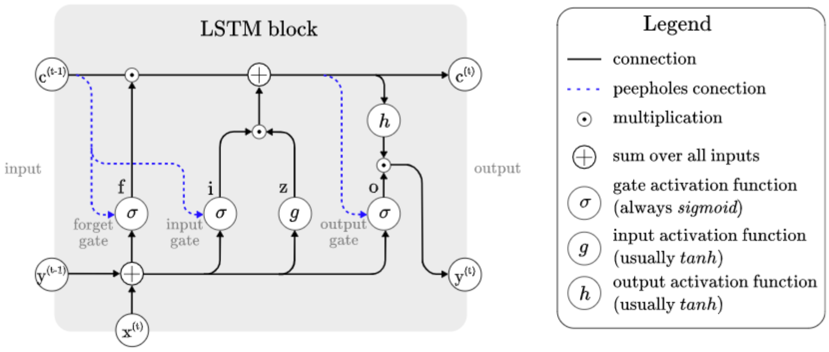


Figure 3.6: The traditional structure of a Vanilla LSTM block, copied from Van Houdt et al. [12]

1. **Forgot Gate (f):** As the name already suggests, this gate re-evaluates the information of the previous cell state, denoted with $c^{(t-1)}$, and determines if some previous information can be discarded based on the previous LSTM's output ($y^{(t-1)}$) and current cell's input (x^t).
2. **Input Gate (i):** This gate updates the information of the previous cell state with some extra relevant information based on the previous LSTM's output and the current input.
3. **Output Gate (o):** Controls whether the output of the current cell becomes visible by protecting relevant information from irrelevant information.

Due to its high applicability and relatively good learning ability, LSTM became a popular model used by high-tech companies. Google, Facebook, and Amazon use LSTM network structures for their translation and speech recognition products [12]. Recently, these models became an interesting research area for time series predictions. An empirical study on financial time series from the stock market by Siami-Namini et al. [47] showed that a state-of-the-art LSTM model outperforms the traditional ARMA model. Chandriah and Naraganahalli [77] predicted the demand for passenger cars in Norway and concluded that a six-layer RNN-LSTM model outperforms the standard used parametric approaches (Croston, SES, SBA, and TSB) in terms of the Mean Squared Errors (MSE) when forecasting lumpy/intermittent demand patterns. The same findings were presented in a comparable study by Jan Gottemeier [6] on forecasting demand for spare parts with lumpy demand patterns for an aviation aftermarket company. The study results concluded that the RMSE of a five-layer LSTM model was approximately 17% lower than the best-performing parametric model, which was an

SES model.

A drawback of the LSTM network is that it contains a large number of parameters, resulting in a complicated system with implementation difficulties. The Gated Recurrent Unit (GRU) network, a recent alternative model proposed by Cho et al. [78], tries to solve this issue by eliminating the output gate; Combining the input with the forget gate into an update gate that determines which information is stored or removed; And adding a reset gate that decides whether to reset previous information for future cell states. Essentially, the GRU is a simplified version of the LSTM with fewer parameters and only two gates instead of three gates.

Multiple studies have compared LSTM networks with GRU models. Ma et al. [66] conducted a case study on spare part order predictions for a Chinese automotive company and concluded that both models showed similar performances, measured by the MAPE, when only considering the demand of the past four consecutive months to predict the demand for the upcoming month. However, the GRU proved to be easier to implement and required less computational time. A similar conclusion was reached by Fu et al. [79] when analyzing the performances of both models, measured by MSE, in predicting five minutes of traffic flow by using traffic sensor information from the past 30 minutes.

3.3.2. Alternative Non-Parametric Approaches

Other viable options for capturing non-linearity in data patterns are in the form of Machine Learning (ML) algorithms. One of the most popular supervised machine learning algorithms for spare part prediction is the Support Vector Machine (SVM) algorithm, also known as the Support Vector Regression algorithm. This algorithm, proposed by Cortes et al. [80], determines an optimal hyperplane that classifies all data points into different classes. The objective of the hyperplane is to maximize the distance between data points of different classes while minimizing the generalization error. All data points are projected in an infinite dimensional space by a kernel function to capture non-linearity [81]. Multiple studies showed that the SVM model can outperform traditional and simple ANN models. However, the same studies also concluded that the SVM may not perform as well when historical data is limited and follows a lumpy or intermittent demand pattern [6, 82–84].

Recently developed transformer networks by Vaswani et al. [85] are gaining interest in the prediction field. These models, currently state-of-the-art in language processing problems, are based on an encoding-decoding structure with a self-attention algorithm that enables the model to compute the input data simultaneously instead of considering the time series' sequential input order. This reduces the computational time significantly compared to RNNs and LSTMs. When it comes to spare part predictions some preliminary findings are looking promising for these models [66, 86]. However, for demand forecasting problems that exhibit lumpy or intermittent demand patterns, more research must be conducted before considering these models as a reliable option.

Another interesting approach, which can either be parametric or non-parametric, is the Bootstrap method. The original bootstrap model generates future demand values purely by randomly selecting previous demand values. Willemain et al. [87] developed a bootstrap method specifically designed to handle intermittent demand patterns. The authors employed a two-state, first-order Markov chain to compute a sequence of demand occurrences. For each non-zero demand occurrence in the sequence, a jittering formula was applied to a randomly selected past demand observation to produce life-like values that may differ from past observations. By summing the entire sequence, the demand value for that specific period was obtained. This process was repeated at least 1000 times to generate a demand distribution for the upcoming period. Comparing their method with traditional time series methods such as Croston, SBA, TSB, and SES, Willemain et al. [87] concluded that their approach achieves higher accuracy. The same method was critically reviewed by Syntetos et al. [88] on jewelry data with different demand patterns and concluded that the bootstrap method outperforms parametric methods when the data is slightly intermittent and has short lead times. However, when the demand is highly intermittent with longer lead times parametric approaches tend to perform better than the bootstrap method. Overall, the bootstrap method is a good alternative when the availability of historical data is limited [30].

3.4. Forecasting Methods' Overview and Final Remarks

Demand patterns of spare parts can provide valuable insights into their characteristics, which can help to improve stock control. To classify these demand patterns, quantitative techniques such as the Syntetos-Boylan-Croston modified Williams scheme are often used. This method categorizes spare parts based on their average demand interval (ADI) and squared correlation of variation (CV^2) into 4 categories: smooth, erratic, intermittent, and lumpy. Aircraft spare parts commonly follow a smooth or intermittent pattern, which is characterized by many periods of zero demand. The uncertain nature of demand intervals and quantities is challenging for forecasting models, and should be considered when selecting an appropriate prediction model. Furthermore, understanding the overall characteristics of spare parts can help in obtaining better predictions. All these variables should be considered as input parameters for forecasting models.

Various methods have been proposed to cope with lumpy and intermittent challenges, including Croston's method. Croston's method is one of the most widely used parametric models in the aviation industry to forecast intermittent and lumpy demand patterns. It decomposes the demand prediction into two separate Single Exponential Smoothing (SES) models: one to predict the demand sizes and another to predict the intervals. However, Syntetos and Boylan [41] showed that Croston's method is biased because it assumes that both the predictions, demand interval and demand size, are independent of each other. To overcome this flaw, a modified version called the Syntetos-Boylan-Approximation (SBA) was proposed. Multiple studies have shown its superiority to the traditional Croston's method. Alternative versions such as the Tuenter-Syntetos-Babai (TSB) model are particularly good at dealing with demand obsolescence.

The above-mentioned parametric approaches have several limitations, including their inability to recognize non-linear patterns which can lead to misinterpretation between the dependent and independent variables. Non-parametric models such as Artificial Neural Networks (ANNs) can overcome these limitations. ANNs are inspired by the human neuron system and consist of multiple connected layers of neurons. These connections are adaptable when new information is introduced. One of the simplest forms of an ANN is a Multi-Layered-Perceptron (MLP) network. Several studies have shown that this model already outperforms the parametric approaches. To include previous information in the training phase of the model, a Recurrent Neural Network (RNN) was proposed. This method has an additional layer that feeds the output into the input layer, and it is trained by a backpropagation (BP) algorithm. Unfortunately, during the training phase, the network can suffer from the vanishing exploding gradient problem. To overcome this, a Long Short Term Memory network was proposed. The neurons in this network contain a memory cell with an input, output and forget gate, which re-evaluates current information and only updates its state with new information when necessary. A modified version of the LSTM networks, named Gated Recurrent Unit (GRU), has recently been proposed. This method simplifies the memory cell of the LSTM by eliminating the output gate, combining the input and forget gate into an update gate, and adding a reset gate. An overview of the above-discussed developments is illustrated in Figure 3.7.

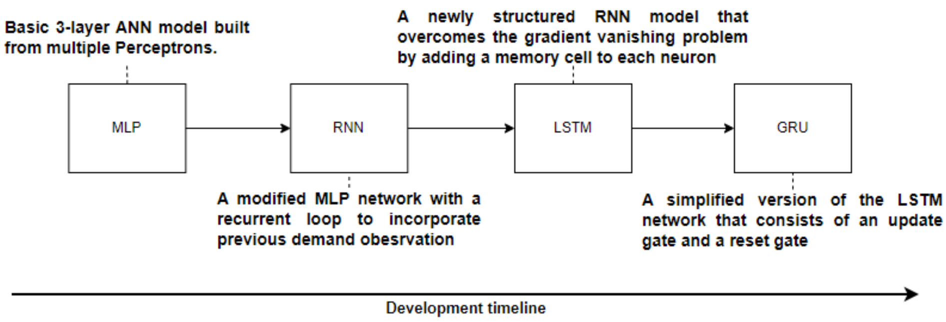


Figure 3.7: The evolution of ANNs used for forecasting intermittent and lumpy demand patterns

In summary, this literature review concludes that the LSTM and GRU networks outperform the other discussed prediction models when dealing with intermittent or lumpy demand patterns. Furthermore, their inner characteristics are beneficial when forecasting multi-variable demand problems. However, it should be noted that deciding which model to use always depends on the dataset characteristics and problem requirements.

4

Customer Segmentation

The purchase behavior of customers depends on many factors. Some customers wait till the latest moment to buy and restore a product, while others may follow a longer-term purchase strategy. In most cases, these purchases are directly correlated with a customer's needs and preferences. With the help of big data technology, customers with similar buying patterns can be grouped together to improve demand predictions. The use of these applications is especially large in the e-commerce sector where all processes are already digitized [89]. This chapter starts with an introduction to customer segmentation in Section 4.1. Followed by Section 4.2 about clustering, A data mining technique used for customer segmentation. Various clustering approaches, along with relevant literature examples, are discussed in this section. Finally, this chapter concludes with a small overview of the discussed subjects and algorithms in Section 4.3.

4.1. Customer Segmentation

An interesting application that arises after the introduction of Big Data is customer segmentation, an approach used in the e-commerce sector to understand, recognize and improve customer behavior by developing marketing strategies dedicated to individual customer groups [90]. Besides the attracted attention in the e-commerce sector, customer segmentation can be a valuable solution that can overcome some drawbacks related to demand forecasting. One of the main challenges in demand forecasting is the limited availability of data, which is necessary to obtain accurate predictions. This limitation is especially problematic during the training phase of ANNs. In some cases, due to the long life cycles of certain products, predicting the next demand cycle may even seem impossible, as previous demand data for specific customers might not be available until a maintenance event occurs for the first time [19]. With the availability of Big Data and mining algorithms, customers with the same buying behavior can be grouped to improve forecasting accuracy [13, 91, 92].

Several options to increase the accuracy of demand predictions with the help of customer segmentation are presented in literature. The initial steps are similar: Instead of considering all the customers at once to train a general forecasting model, the customers are first divided into smaller subsets. Kalchschmidt et al. [13] developed a forecasting model for each customers group separately and combined all groups' results to predict the total demand, as shown in Figure 4.1. While Caniato et al. [92] only forecasted demand for the customers closest to the centroid of the cluster and extrapolated those results to the entire group to have an overall forecast. The latter one is only of interest if the density of customer groups is small and there is enough previous demand data of the most centralized customer available.

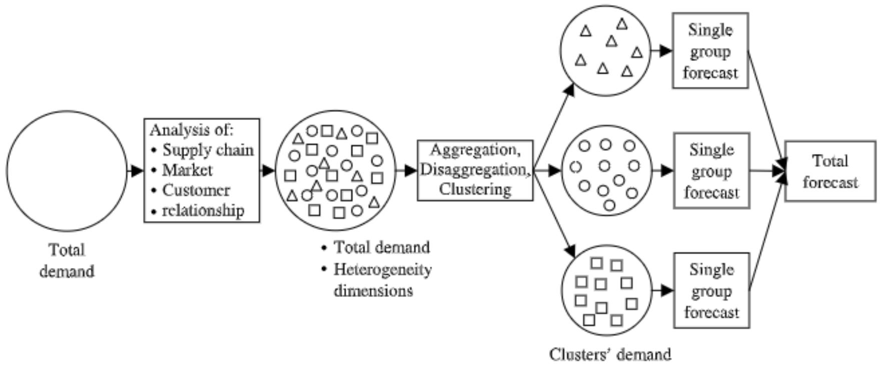


Figure 4.1: Customer segmentation example methodology proposed by Kalchschmidt et al. [13] to increase forecast accuracy

The fundamental idea behind customer segmentation is to group customers based on similarities in their buying behavior. Therefore, it is crucial to explore and understand the drivers behind purchases. According to Caniato et al. [92], there are three main variables influencing demand: Systematic variability, managerial variability, and randomness. Systematic variability includes variables related to products and overall customer preferences. These variables represent the entire industry and are easy to explore via historical sales data sets. Examples are seasonality and trends. Managerial variables, on the other hand, cannot be captured by only analyzing historical demand data. Inner characteristics of the customer are needed to explain certain behavior. These variables represent the business strategies and management decisions of the buyer and seller. Typical examples of managerial variables are promotional activities and customer strategies/policies. Lastly, there are random variables. These variables are useless for segmentation as they are different for each customer and purely depend on individual needs. Caniato et al. [92] concluded that systematic and managerial variables are positively correlated with demand patterns, and although they performed a case study for a specific company of a specific industry, they expect similar results for other industries.

A popular method to describe customers' preferences and transaction activities is the RFM model, developed by Hughes [93]. This method focuses on three systematic customer variables: Recency (R), Frequency (F), and Monetary (M). Recency equals the interval length since the last purchase. The shorter this length, the larger the R-value. The number of unique transactions in a predetermined period is expressed by the Frequency, and the corresponding amount of money spent during those transactions equals the Monetary value. Many studies have used this method to cluster customers on their behavior. Wang et al. [90] used the RFM model to cluster customers into Favorite, General, and Inactive customers. Over the years, many modifications and extra variables were proposed to improve the method. Güçdemir and Selim [94] added five extra variables describing the developing relationship with the customer since their first purchase: (1) Loyalty, the time since the first purchase; (2) Average annual demand; (5) Relationship potential computed by multiplying loyalty with recency; (4) Average percentage change in annual demand; And (5) Average percentage in annual sales revenue. Their analysis showed that loyalty, frequency, and the percentage change in annual demand and sales revenue were important segmentation factors.

Airworthiness and safety are important drivers in the aviation sector, as explained in chapter 2. These are similar for all customers. However, maintenance and operating strategies differ per customer and probably influence their buying behavior. Therefore, it is interesting to identify these managerial variables for customer segmentation to improve forecasting accuracy.

4.2. Clustering

The data mining technique for customer segmentation is clustering, an unsupervised learning approach that organizes unstructured unlabeled data sets into smaller subsets named clusters. The algorithm forms clusters of data points, with higher similarities among each other than among other groups [95, 96]. It should be noted that these techniques only focus on reassigning data points and do not specify the importance of each cluster [94]. Different types of clustering algorithms are discussed in literature. The most common techniques can be categorized into Hierarchical or Partition clustering techniques [91].

The concept of hierarchical clustering is to construct a hierarchy of clusters step-by-step, which can be done via an agglomerative or divisive approach. The agglomerative approach, also known as the bottom-up approach, initially expresses all data points as individual clusters. At every time step, the most similar clusters are merged until an optimum solution is reached. The divisive approach, also known as the top-down approach, works the other way around. Initially, all data points are together in one cluster, and at every step, the most diverse data points are split into smaller clusters until an optimum solution is reached.

Popular hierarchical clustering techniques are single linkage, complete linkage, and Ward's method. Single linkage combines two separate clusters by using the smallest distance between the closest and most similar data points. While in complete linkage, the shortest distance between the farthest, most dissimilar data points is used to decide which clusters can be combined. Wards method works differently and computes the minimum sum of squared errors to determine which clusters to merge or split [14, 94]. The advantages of Hierarchical clustering techniques are their easy and straightforward implementation and the self-determination of the optimal number of end clusters throughout the hierarchical structure. However, the downsides of this technique are the long computational time and their inability to recover after a time step, which makes them infeasible for large datasets [97].

Partitioning clustering techniques have a different approach than hierarchical techniques. Instead of starting at the bottom or top, all data points are initially divided over a predetermined amount of clusters and reassigned until an optimum criterion is met. The initial partition (random or nonrandom), reassigned procedure, and stop-criteria depend on the chosen algorithm [90, 98]. Empirical studies have shown that partitioning clustering techniques tend to perform better and quicker than hierarchical techniques when the dataset is large, and the initial starting conditions are known [91, 98]. The most commonly used algorithm is K-means partitioning which is further explained below next to some other high-performance clustering techniques.

4.2.1. K-Means

The most popularly used clustering method is the K-means method. Since this is a partitioning method, all data observations are initially split into a predetermined number of clusters, denoted as K clusters. The algorithm then computes the distance between the data observations and centroids of each cluster. At every step, the algorithm minimizes the squared distance by reassigning data points to other clusters, after which it recalculates the centroid of the newly formed clusters. This assignment procedure is repeated until no reassignments occur [91, 94]. The easy implementation, high overall accuracy, and the ability to handle large datasets in a short amount of time are advantages of the K-means algorithm. On the other hand, the biggest drawbacks are the predetermination of final K clusters and the initial starting point, which strongly affects the final outcome. Furthermore, the algorithm is sensitive to outliers, and the obtained results will completely change when the dataset is rescaled [97, 99, 100].

Several studies have shown the practicality and performance accuracy of the K-Means method. Murray et al. [91] used this algorithm to cluster customers from a material supplier into 10 different segments based on seasonality, industry type, etc. They concluded that outliers in the data had a bigger effect on the outcome than using other clustering techniques. Similarly, in a study conducted by Güçdemir and Selim [94], the K-means algorithm outperformed hierarchical clustering methods in terms of the squared sum of distances within clusters. The study focused on segmenting customers of an international OEM using variables derived from the RFM model. The K-means method successfully clustered the customers into four distinct groups: best, valuable, average, and potential customers.

4.2.2. Self Organising Feature Maps (SOFM)

Another widely used algorithm for clustering problems is the Self Organising Feature Maps (SOFM) developed by Kohonen [101]. SOFM are unsupervised learning ANNs that transform high-dimensional inputs into a low-dimensional topology while preserving the essential data characteristics as much as possible. The same fundamental ANN principles, explained in Subsection 3.3.1, apply to this network. The overall structure of the SOFM network, which includes an input and output layer, is illustrated in Figure 4.2. In the input layer, each data observation is connected via adaptable weights to every neuron in the output layer. The neurons in the output layer are, on their side, connected laterally with each other.

During the training phase of the algorithm, the input data points are projected into the low-dimensional grid via the output neurons. As a result of the interconnected output neurons, high-density areas start to occur, representing similarities between data points [102, 103]. An example of cluster determination via SOFM from Seyedan et al. [14] is presented in Figure 4.3.

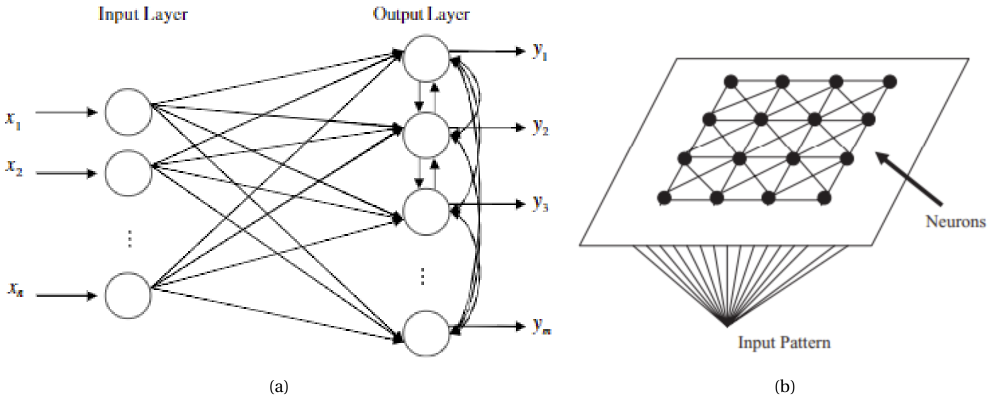


Figure 4.2: Simplematic sketch of the SOMF network structure

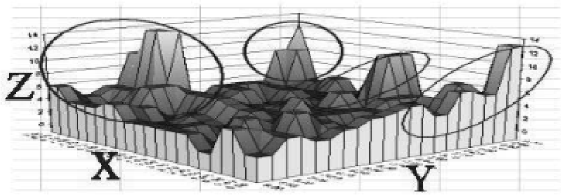


Figure 4.3: Determination of the optimal number of clusters via SOFM [14]

Positive aspects of SOFMs are their ability to handle large datasets and, contrary to the K-means method, they do not require the predetermination of optimal clusters due to their unsupervised self-organizing characteristic [91]. However, the complexity of the training set-up and different parameters can significantly affect the outcome. Furthermore, similarly to ANNs, they require a relatively long computation time [99].

Kiang et al. [104] utilized a SOFM network to cluster customers of an American Telecom company. By including demographic variables and additional customer attitude information concerning long-distance communication, they concluded that SOFM network outperformed the traditional K-means method in generating more precise clusters. The accuracy performance between the two methods was measured by the total within-cluster variance.

another study conducted by Sanchez et al. [105], employed a SOFM to classify customers based on their hourly energy consumption. The research focused on clustering the current energy users of a Spanish energy company, aiming to predict the energy consumption of future customers. The results showed that the SOFM successfully identified 10 distinct clusters and thereby provided valuable information on residential users within the Spanish electricity market. Unfortunately, their method was not compared with other clustering approaches, making it difficult to draw definitive conclusions regarding the superiority of the SOFM network over other approaches.

4.2.3. Genetic Algorithms (GA)

The application of Genetic Algorithms (GA) to solve clustering problems is widely studied. Originally, These algorithms were designed for optimization and search problems. Their good search capabilities make them an interesting alternative solution for clustering big data sets [106]. GAs are inspired by biological evolution and natural genetic principles. The parameters are represented as a population of multiple chromosomes. Many variations are developed over the years, some more complex than others, all with specific capabilities.

However, the fundamental principles are similar. The main terms and processes of a simple clustering GA for market segmentation are explained below [107]:

1. **Initializing of a population:** A population of chromosomes equal to the size of the data set is created at the start of the genetic algorithm. Each chromosome represents the centroids of the predetermined clusters. For every chromosome, these centroids are obtained by randomly selecting a data point from the data set. Thus, a data point is initially considered as the center of a cluster during the initialization of the GA [107].
2. **Fitness computation:** Per chromosome, all remaining data points are assigned to the nearest cluster. After which, the original clusters' centroids are updated with their respective mean points. Finally, the clustering metric for every chromosome in the population is computed. Various clustering metrics exist. An example is the Euclidian distance which equals the total distance between the data points and their respective clusters' centroids. This metric illustrates the fitness of the chromosome [107].
3. **Parent selection:** Different selection techniques exist for GA, all with their own characteristics. However, the main principle is the same: Survival of the Fittest. A mating pool containing the best chromosomes is created for further genetic operations. This is commonly done by creating duplicates of each chromosome, proportional to their early determined fitness in the population [107].
4. **Crossover determination:** Two 'parent' chromosomes are selected from the mating pool to create two new 'child' chromosomes. This process is called the crossover and can be done in many ways. A straightforward solution is to randomly determine a crossover point to split the chromosome into a left and right part. By exchanging the right parts with each other, two chromosomes are created [107].
5. **Mutation:** The centroids in the chromosome are slightly mutated to create unique chromosomes. Again, various mutation approaches exist. Normally, an uniformly distributed variable between 0 and 1 is randomly added or subtracted from the centroid [107].
6. **Termination:** The process is repeated until a termination criterion is met. After each iteration, the chromosome with the highest fitness is preserved. After termination, this chromosome contains the 'optimal' centroids of the clusters [107].

Similarly to the K-means clustering technique, the number of final clusters needs to be determined in advance, which is not ideal for most problems. Another disadvantage relates to the increased complexity of the algorithm compared to traditional methods. However, the performance accuracy of GAs tend to be better than the simpler K-means methods and hierarchical approaches. Cowgill et al. [108] concluded that the GA outperformed the K-means and Ward's method in terms of the Euclidian distances when applied to artificial datasets with pre-defined cluster groups. The same observation was made by Maulik and Bandyopadhyay [107], who compared the results of GA and K-means on multiple datasets, including artificial and real-life datasets such as sound vowel data, iris/pupil data, and crude oil data. The results consistently demonstrated that the GA significantly outperforms the K-means method across all datasets in terms of the Euclidian distances.

Kim and Ahn [109] utilized a GA for an online shopping market recommendation system to categorize online users into five distinct clusters based on their online opinions and preferences. The results indicated that the GA clustering approach led to better segmentation compared to the k-means method and SOFM methods. Nevertheless, the study acknowledged that other effective error metrics to accurately compare clustering algorithms should first be considered or applied before making a definitive conclusion.

Liu and Ong [110] emphasized the significance of carefully selecting relevant variables when conducting market segmentation to avoid distorting the clustering algorithm. The authors applied a GA to identify the most promising variables for cluster determination, followed by a traditional k-means method to cluster the customers. The GA algorithm was applied to a German Credit dataset containing 20 distinct features, such as historical credit records, age, gender, marital status, current job, and other personal information. Their analysis concluded that the GA effectively rejected the irrelevant variables, resulting in better distinct clustering groups compared to an all-variables included K-means method.

Traditionally, GAs were developed for solving single-objective optimization problems. Recently, multi-objective GAs were proposed to increase the accuracy of the final cluster outcomes. Two objective functions are simultaneously optimized to capture all the dataset characteristics and achieve a better, more balanced result. In the case of a clustering problem, the fitness of a chromosome may not only dependent on the similarity of data points within a cluster but also on the dissimilarity between data points from other clusters [96, 111]. For instance, Ben Ncir et al. [112] developed a multi-objective GA to segment customers of a retail bank by incorporating socio-demographic and behavioristic variables. Their results concluded that the multi-objective approach outperformed single-objective methods, such as K-means, when analyzing the within-cluster variance and the dissimilarities between the nearest clusters.

4.2.4. Alternative Hybrid methods

Alternative hybrid approaches are proposed in literature to overcome the shortcoming of single clustering techniques. A two-stage method, containing Ward's Hierarchical method and K-means partitioning algorithm, was suggested by Punj and Stewart [98] for market segmentation to solve K-means issue related to the prior determination of end clusters. This solution overcomes both methods' drawbacks and ends up with a combination of their best features. First, Ward's method determines the optimal number of clusters and starting points, which are necessary to initialize the K-means method. After which, the k-means method assigns the data points to the clusters, since partitioning methods tend to outperform hierarchical methods when the necessary starting information is available. To overcome the non-recovery characteristic of hierarchical methods, Kuo et al. [99] proposed to replace Ward's method with SOFM in the earlier suggested two-stage methodology, as SOFM can rearrange data points to the nearest cluster. The prior information of final clusters and starting points necessary for the K-means algorithm can easily be determined by sight, as shown in Figure 4.3. Their study results on multiple customer datasets showed that the SOFM plus K-means method slightly outperforms the Ward's plus K-means two-stage approach. In a subsequent research, The same researchers proposed a similar two-stage approach with a GA instead of the simpler K-means method. After comparing all methods, they concluded that the SOFM followed by GA is superior to the other approaches Kuo et al. [106].

Another proposed approach to enhance clustering is by incorporating the principles of the Fuzzy set theory into an existing technique. Fuzzy logic reflects human thinking and its corresponding uncertainty by allowing data points to belong to multiple clusters instead of being restricted to just one. A membership grade is assigned to each data point to indicate its potential presence in a cluster. Fuzzy clustering is especially interesting when clusters may overlap in a data set. Commonly used fuzzy clustering methods are Fuzzy C-means and Fuzzy K-means. [91, 94]. For instance, a study conducted by Ansari and Riasi [113] focused on clustering customers of a steel company using a combination of Fuzzy C-means and a GA. The clustering approach aimed to cluster customers into two groups based on variables derived from the RFM model. The algorithm successfully identified the two clusters: one including loyal customers and the other consisting of more recent/newer customers. The results indicated that the combined approach obtained a lower MSE score compared to using the algorithms individually.

Many literature reviews state that there is not a particular technique that would always outperform other models since it heavily depends on the data set characteristics [114]. Therefore, Seyedan et al. [14] proposed to use an ensemble learning technique to combine the outcome of multiple clustering methods into one final solution to forecast demand. The cluster results of the K-means method and three hierarchical clustering techniques were combined via Majority voting. This algorithm assigns data points to their respective clusters based on the overlapping results of all clustering methods. The final obtained segments were then used as input variables for an LSTM and prophet model to predict future demand. Results show that the clusters generated via majority voting achieve better forecasts than when the methods were used individually.

4.3. Conclusion

Big data techniques have been widely adopted across many industries in recent years, particularly in the e-commerce sector where everything is digitized. A popular Big Data application is customer segmentation, which is an unsupervised classification technique that groups customers with similar characteristics. Several studies have suggested that clustering customers based on their buying behavior can improve forecasting accuracy.

Various models have been considered to cluster customers based on their inner characteristics and purchase behavior. The K-means method is widely used in many applications and assigns data points to the nearest cluster by calculating the respective distances toward the clusters' centroids. Another considered approach to cluster customers is using Genetic algorithms (GAs) which are inspired by biological evolution and natural genetic principles and known for their good search capabilities. A GA typically consists of a population of chromosomes that are paired and mutated in order to find an optimal solution. GAs have the ability to optimize multiple objectives, whereas the K-means method only optimizes one. However, both methods require the number of final clusters upfront, which can be challenging to determine. The Self Organising Feature Map (SOFM) network is the third approach that is considered to cluster the customers and has an unsupervised self-organizing feature that independently determines the final amount of clusters. The model projects high-dimensional inputs into a low-dimensional topology via output neurons, which results in high area densities that are equal to the final cluster. The disadvantages of SOMF are its complexity and long computational time.

A combination of two models is preferred to overcome the particular model drawback and end up with the best results. SOFM is used to determine the number of final clusters, and the results are then utilized by a GA to group customers. Variables from the RFM model should be considered during the clustering process. Other interesting variables that can express customer behavior in the aviation sector include flight schedules, the number of available aircraft, aircraft type, airline strategy, and its operating region. The final customer segmentation can be used as an input feature for the forecasting model to improve its accuracy.

5
Pattern Mining

This chapter discusses relevant pattern mining techniques to evaluate and discover correlations between items in a data set. Section 5.1 starts off with a detailed explanation of Association Rule Mining. After which, two powerful ARM algorithms are described. Another mining technique, Sequential Pattern Mining, is discussed in Section 5.2. The chapter concludes with an overview of both techniques and their potential usage for spare part predictions in Section 5.3.

5.1. Association Rule Mining (ARM)

The overlapping term for discovering correlations and patterns in data sets, which are understandable and helpful for decision-making processes, is pattern mining. ARM, short for Association Rule mining, is the best-known Big Data pattern mining technique. This technique aims to discover relations between together ordered items by analyzing all transaction records in a data set [114]. Agrawal et al. [115] were the first to formulate this problem and referred to it as the market-basket analysis. In their problem statement, they mentioned that a transaction does not necessarily have to consist of items that were bought together but could also consist of multiple items, purchased over a longer period. This property makes ARM a powerful tool for different industry types.

ARM techniques describe the correlation between items via association rules. Each association rule consists of a condition/antecedent part and a prediction/consequent part and is formulated as an if-statement: **If** product A and product B are bought **THEN** product C is also bought. If the conditional part, in this example product A and product B, is true, then product C is also true. The mathematical expression is given in Equation 5.1. Usually, The condition part contains multiple items while the prediction part only represents a single item[114].

$$AB \Rightarrow C \tag{5.1}$$

Different ARM techniques have been proposed since their discovery. All with unique characteristics, depending on specific data set properties. In general, these algorithms can be classified into three categories: Categorical, numeric, and fuzzy rules. Most algorithms are designed to generate categorical association rules. These are binary rules that only specify the presence of items in a transaction and do not mention anything concerning their corresponding quantities. Whereas numeric association rules describe these quantities as boundaries. However, these boundaries are represented as intervals and, therefore, unable to describe smooth changes between the interval boundary layers, which often occurs in practice. Fuzzy association rules overcome this limitation by representing the quantities as fuzzy sets [111, 114]. This literature review mainly focuses on algorithms computing categorical association rules since the main interest of this case study relates to the determination of correlation between products in sale records. First, the oldest and most commonly used ARM technique, the A-priori algorithm, is described in Subsection 5.1.1. After which, an alternative approach in the form of genetic algorithms is discussed in Subsection 5.1.2.

5.1.1. A-Priori Algorithm

The A-Priori algorithm, developed by Agrawal and Srikant [116], was the first technique to obtain relevant categorical association rules in large data sets. The algorithm first determines the frequency of all itemsets in the sales records by computing its support value. The items with a higher support value than the predetermined minimum support level are labeled as large itemsets, while the remaining itemsets are labeled as small itemsets. The second part of the algorithm generates the association rules from the large itemsets by computing the confidence value, which describes how often a rule is true. If the confidence value is above a certain threshold, the association rule holds. The formulas to calculate the support and confidence of itemset ABC are given in Equation 5.2 and Equation 5.3, respectively. When both the support and confidence surpass the minimum required level, the rule $AB \Rightarrow C$ holds.

$$Support = \frac{frequency(ABC)}{N}, \quad Support \geq Support_{min} \tag{5.2}$$

$$Confidence = \frac{frequency(ABC)}{frequency(AB)}, \quad Confidence \geq Confidence_{min} \tag{5.3}$$

Determining the minimum support level is challenging. Low support levels generate many rules, which can be difficult to interpret or distinguish. However, these rules can still contain valuable information that is relevant to certain company processes. Whereas, this information wouldn't be available and probably be unaware of when the support threshold is set too high [114]. Another challenge of the algorithm is regarding the two-stage structure. To determine the large itemsets, the entire database has to be scanned for every set separately, which is a time-consuming process that exponentially increases with the size of the database. Furthermore, the generated association rules by the algorithm are only based on frequent appearances in a data set. Other factors that could be of interest when determining a rule's relevancy, such as comprehensibility and interestingness, are thus not included in the algorithm [111]. Nevertheless, the A-Priori technique is still commonly used by many companies and known as a proven solution for mining association rules.

Chen and Wu [117] utilized the A-priori algorithm to uncover associations between different orders, which could increase order/batch-picking efficiency in warehouses. Although, the researcher did not compare the A-priori algorithm with other ARM approaches, they concluded that the A-priori algorithm effectively improved the efficiency of order batching problems in warehouses. In another study by Magdalene Delighta Angeline [118], the A-priori algorithm was employed to identify relevant connections among variables describing student educational information, including class attendance, delivering assignments, and more. The objective of the research was to identify the good, average, and poor-performing students. The study successfully derived high-value rules by considering the support, confidence, and lift metrics.

5.1.2. Multi Objective GA for Association Mining

An alternative approach with Genetic Algorithms (GA) is suggested by Dhaenens and Jourdan [114] and Mukhopadhyay et al. [111], to overcome the challenges/drawbacks of the A-Priori algorithm. As previously described in Subsection 4.2.3, GAs are based on the principles of biological evolution and are highly effective in searching databases. Instead of scanning the database for every item set separately, GAs skip the support phase and generate the rules directly. This approach decreases the computational time significantly and can optimize multiple objectives. The algorithm's main components for ARM are similar to the already described clustering GA and still consist of: population initialization, fitness determination, crossover, mutation, and termination. The main differences relate to the generation of chromosomes and their fitness computation.

Two main design approaches can be followed when using GAs for ARM: The Michigan design or the Pittsburgh design. The Michigan approach is better suited for ARM since every solution describes a rule. Whereas the solutions generated by the Pittsburgh design describe a possible set of rules, which are normally difficult to interpret. Following the Michigan approach, the chromosomes can either be represented as binary strings or integers. Binary chromosomes are built out of bits. Each item is presented as a pair of bits, describing its presence or absence in the respective rule. There are four possibilities to express the state of an item: 00, 11, 01, and 10. Items that are part of the rule are denoted with 00 or 11, depending on their presence in the condition or prediction part of the rule, respectively. The other pairs, 01 and 10, describe the absence of the items in the respective association rule. An advantage of binary representation is that all items are present in a chromosome, meaning that each chromosome has the same length, equal to 2 times the available items

[111]. An example of binary chromosome representation for a data set of 4 items (A, B, C, and D) is given in Equation 5.4. In this example, rule $AB \Rightarrow C$ holds. It is worth mentioning that a programmer can choose a different binary bit representation. Some studies express the presence of items as 01 and 10 instead of 00 and 11 [111].

$$Binary = \{00|00|11|01\} \tag{5.4}$$

$$Integer = \{2|ABC\} \tag{5.5}$$

The entire data set has to be transformed into a binary set to create binary chromosomes, which can be challenging. Furthermore, binary chromosomes tend to have an extremely long length because all items are present. Representing the chromosomes as integers overcomes this issue. An example of an integer chromosome representation is given in Equation 5.5. The first gene describes the position between the condition and prediction part of the rule. All items on the left of the indicated position belong to the condition part, and all items on the right belong to the prediction part. The problem with integer chromosome representation is its variable length, which introduces inconsistencies and difficulties during crossover and mutation. Special operators need to be selected to guarantee that the algorithm runs smoothly. Whereas standard operators are already effective when using binary chromosome representations [111].

A benefit when using GAs for ARM is the ability to optimize multiple objectives. Instead of maximizing the support and confidence of rules, the quality of the rules can also be obtained [114]. According to Mukhopadhyay et al. [111], several studies proposed additional metrics. Interestingness to describe the rule's quality and comprehensibility to express its complexity/interpretation level are commonly used. The interested reader may refer to the study of Mukhopadhyay et al. [111] for a complete overview of multi-objective GAs. Furthermore, its worth mentioning that GAs are also applicable to numeric and fuzzy ARM. The respective quantities are then incorporated into the chromosomes, which increases the complexity of the GA.

Wakabi-Waiswa and Baryamureeba [119] developed a multi-objective GA to uncover meaningful relationships among 18 animal attributes. These attributes included characteristics like the presence of hair, legs, backbone, egg-laying ability, milk production, and more. All these attributes were collected in an extensive real-world Zoo dataset consisting of over 100 different animal species. The GA was specifically designed to optimize three distinct objectives: prediction accuracy, comprehensibility, and J-measure. The study findings demonstrated that the algorithm successfully discovered precise rules, some with more than 90% accuracy.

5.2. Sequential Pattern Mining

Another mining technique that discovers interesting patterns in big data sets is sequential pattern mining. The difference with ARM is that this technique considers the sequential ordering process by analyzing sequences of purchases. Next to sequential databases, this mining technique can also be used to discover patterns in time series. However, this literature review will only reflect on the mining process for sequential databases, as the eventual database consists of transaction records. Similar to ARM, sequential pattern mining techniques are searching for itemsets, in this case, subsequences from a sequence data set, with a higher support and confidence value than the predefined thresholds. Various sequential pattern mining algorithms and improvement techniques have been proposed in literature, since its introduction by Agrawal and Srikant [120]. To a large extent, these techniques depend on the chosen search algorithm, sequential database representation, possible sequential pattern generation, and each sequence support determination [15]. First, different search algorithms are discussed in Subsection 5.2.1. After which, the database representation and their respective properties, including pattern generation and support computation, are discussed in Subsection 5.2.2. This section ends with an overview of the available algorithms in Subsection 5.2.3.

5.2.1. Search algorithms

A search algorithm creates an entire database of possible sequences from available transaction records by performing s-extensions and i-extensions. These two straightforward sequence generation techniques append an extra item to an already generated sequence. An i-extension operation adds a new item to the last transaction of a sequence, which means that the appended item was ordered during the same transaction as the previous item in the sequence. An s-extension operation works slightly differently as the extra item joins the sequence as a subsequence, meaning that the appended item was purchased at a later transaction moment than the last items in the sequence. To better understand the difference between these two extension operations, the reader may refer to the second and third steps of the breadth-first algorithm, illustrated in Figure 5.1. In general, three different search algorithms for sequential pattern mining are available: The breadth-first search algorithm, the depth-first search algorithm, and the pattern-grown algorithm. All three techniques are constantly performing i-extensions and s-extensions. However, their problem-solving approach is different [15].

⟨{a}⟩	⟨{a},{a}⟩, ⟨{a},{b}⟩, ⟨{a},{c}⟩	⟨{a,b}⟩, ⟨{a,c}⟩
(1) Starting with a 1-sequence	(2) Performing **s-extension** operations	(3) Performing **i-extension** operations

Figure 5.1: The first generated sequences of the breadth-first search algorithm [15].

The breadth-first algorithm, also known as the level-wise approach, starts with collecting frequent sequential patterns of 1 item from the transaction records. By performing s-extensions and i-extensions on these collected '1-sequences', '2-sequences' are created. After which, '3-sequences' are generated from the '2-sequences', and so on. The process terminates when no new sequences can be formed. Alternatively, the depth-first search algorithm can be used to create a sequential database. This algorithm is relatively similar to the breadth-first algorithm. However, i-extensions and s-extensions are performed first to extend the selected '1-sequence' with new items until all possible sequences are generated. After which, the whole process is repeated with the next selected '1-sequence'. The first few generated sequences of the breadth-first and depth-first search algorithms are shown in Figure 5.1 and Figure 5.2, respectively.

⟨{a}⟩	⟨{a,b}⟩, ⟨{a,b,c}⟩, ⟨{a,c}⟩	⟨{a},{a}⟩, ⟨{a},{a,b}⟩, ⟨{a},{a,c}⟩
(1) Starting with a 1-sequence	(2) Performing **i-extension** operations	(3) Performing **s-extension** operations

Figure 5.2: The first generated sequences of the depth-first search algorithm [15].

The problem with these two search algorithms is that they both create extremely large databases with many possible sub-sequences, which can cause difficulties during the rule's determination phase. The downward-closure property, sometimes referred to as the Apriori property or anti-monotonicity property, partially solves this issue by reducing the search space. This property only explores extended sequences in the generated database when its previous version (the sequence before the i-extension or s-extension) has a higher support value than the predefined threshold, because an extended sequence cannot have a higher support value than its predecessor. To conclude, with the help of the downward-closure property, the database doesn't have to be scanned for every sequence to determine its support and confidence, which decreases the computational time [15].

Both algorithms, the breadth-first search and the depth-first search, create a database of possible sequences from the obtained '1-sequences' of the transaction records. The transaction records are thus only scanned once, which is beneficial for the computational time. However, it is possible that non-existing candidate sequences, sequences that do not appear in the transaction records, are created when applying these algorithms. Pattern-growth algorithms overcome this issue by scanning the transaction records after every extension to evaluate its existence. This process increases the computational time. To reduce the computational time, the algorithm creates a so-called projected database for the currently evaluated sequence, which only consists of items that appear after this sequence in the transaction records. The support value of each item in the projected database is then calculated to make sure that the downward-closure property holds. After which, the depth-first search algorithm extends the sequences with the possible items in the projected database by performing i-extension and s-extensions. A drawback of this technique is that generating an additional projected database increases the memory consumption, which is not always preferable or even an option [15].

5.2.2. Database representation & optimization techniques

Next to the different search algorithms, there are various ways to represent the generated sequential database. Traditionally, a horizontal database representation was used, where all possible sequences are summarized under each other and indicated by a unique sequence identifier (SID). Newer algorithms that make use of the depth-first search technique store the generated sequences in a vertical database. This vertical representation indicates the IDList of every item, which summarizes all possible positions of an item in every generated sequence. A simple example of a horizontal and vertical database representation is given in Figure 5.3a and Figure 5.3b, respectively. Horizontal databases can be transformed into vertical databases and vice versa.

SID	Sequence
1	$\langle\{a,b\},\{c\},\{f,g\},\{g\},\{e\}\rangle$
2	$\langle\{a,d\},\{c\},\{b\},\{a,b,e,f\}\rangle$
3	$\langle\{a\},\{b\},\{f,g\},\{e\}$
4	$\langle\{b\},\{f,g\}\rangle$

(a) Horizontal database

a		b	
SID	Itemsets	SID	Itemsets
1	1	1	1
2	1,4	2	3,4
3	1	3	2
4		4	1

(b) Vertical representation of item a and b

Item x	IDList of x as a bit vector
a	100001001100000
b	100000011010010

(c) Bit vector representation of item a and b

Figure 5.3: Evolution of the sequential database representation [15].

Vertical databases have a few specific attributes that make them powerful for sequential pattern mining. First, an item's/sequence's support value can easily be determined by counting the number of distinct identifiers in the corresponding IDList. This property decreases the computational time compared to the horizontal database representation, where the support values are calculated by scanning the transaction records for every item or sequence individually. The second attribute corresponds to the creation of new IDLists. Vertical databases can create an IDList for an extended sequence by comparing the IDLists of its components. Again, this means that the original database doesn't have to be scanned to create a new sequence, which makes them extremely efficient. The vertical database representation becomes even more efficient when transforming the IDList structures into bit vectors. These vector representations reduce the overall memory consumption as they represent the entire IDList structure as a single line [15]. The explained evolution of the database representation is illustrated inFigure 5.3.

5.2.3. Available Algorithms

Every improvement step described above emerged from a newer developed algorithm. One of the first algorithms, GSP, applied the breadth-first search technique to create a horizontal database. Due to the earlier described limitations of these properties, the Spade algorithm was proposed. This algorithm utilizes a depth-first search algorithm to generate a vertical database. As the general memory consumption increased, bit vectors were introduced, which led to the introduction of the BitSpade algorithm. A relatively similar technique is the Spam algorithm. A recent improvement of the Spam algorithm is the CM-Spade algorithm and of BitSpade algorithm is the CM-Bitspade algorithm. These algorithms generate a co-occurrence MAP (CMAP) that contains all possible frequent 2-sequence item sets. If the last two items of a sequence are not in the CMAP, the sequence can be ignored for further operations. This extra step should prevent generating and analyzing infrequent sequences and reduce the computational time [15]. According to a recent survey on sequential pattern mining techniques by Fournier-Viger et al. [15], the CM-Spade algorithm outperforms all other available algorithms.

5.3. Conclusion & Applicability

Association Rule Mining (ARM) and Sequential Pattern Mining algorithms are powerful techniques to discover relations between items/products in big data sets. ARM algorithms describe correlations between items within a single transaction without taking the sequential order of the transactions into account. Sequential pattern mining algorithms, on the other hand, specify these sequences. Using a Genetic Algorithm to determine association rules overcomes the challenges of traditional algorithms. Instead of first searching for frequent itemsets, GAs directly starts with obtaining the association rule. Furthermore, they can optimize multiple objectives simultaneously to capture more specific associations. The CM-Spade algorithm is the most advanced algorithm to discover sequential sales record patterns. This method is an improved version of the earlier proposed Spade method with additional features to reduce memory consumption and run time.

An aviation aftermarket company can obtain valuable information from its sales records by utilizing these two methods. Which mining technique is more applicable depends on the transaction database. It is suspected that most spare parts are bought independently during separate transactions. With this knowledge, one can assume that sequential pattern mining techniques are better suited for finding correlations between spare parts and their respective maintenance events. However, by combining multiple transaction moments into one basket, ARM should also be capable of finding the relation between spare parts. The discovered correlation between spare parts can be used to improve forecast accuracy and increase customer services.

6

Conclusion & Future scope

The conclusion and future scope of this research are stated in this chapter. First, a conclusion on the main findings is presented in Section 6.1. After which, the main research questions and a first preliminary future scope for the remaining part of the project are described in Section 6.2

6.1. Conclusion

The main objective of this study was to provide a comprehensive overview of spare parts demand forecasting techniques that apply to the aviation aftermarket industry. Furthermore, extra research on improvement strategies to increase prediction accuracy was necessary. In general, it can be concluded that the prediction of spare parts demand is challenging due to strict regulations on aircraft maintenance, the variation in maintenance processes, the various spare part characteristics, high variation in competing stakeholders, many different customer characteristics, and lumpy/intermittent demand patterns. All these characteristics have been considered while analyzing different forecasting models.

It is concluded that non-parametric forecasting models are superior to traditional parametric approaches, due to their ability to capture non-linear demand patterns. various non-parametric models were analyzed on their performances. The Long Short-Term Memory (LSTM) network tends to outperform other alternative artificial neural networks (ANNs). The neurons in the network consist of a memory cell with an input, output, and forget gate, which re-evaluates all incoming information and only updates its state when new information is interesting. This property overcomes the vanishing gradient problem of Recurrent Neural Networks (RNNs). The Gated Recurrent Unit (GRU), a recently developed modified version of the LSTM, is shown to perform similarly to the LSTM.

Two big data techniques were discussed to improve the accuracy of the prediction model. The first technique involves using a clustering algorithm to group customers based on their buying behavior, which should increase the amount of accurate data that is necessary for training the prediction model. First, a clustering algorithm that clusters customers based on their buying behavior is proposed. This would generate accurate training data for the LSTM model. Various clustering models have been considered. From conducted research, it is concluded that a Genetic Algorithm (GA) outperforms the traditional K-means method. However, both methods require the number of final clusters as input variables. To overcome this drawback, a two-stage method that includes the Self-Organizing Feature Map (SOFM) is suggested.

The second big data technique that was investigated is pattern mining, which can be done using either association rule mining (ARM) or sequential pattern mining. A GA is suggested for ARM as it optimizes multiple objectives and decreases the computational time by skipping the support determination of every rule. The best algorithm for sequential pattern mining is CM-Spade, which incorporates additional features to further reduce computational time.

6.2. Research Question(s) & Future outline

This literature review serves as the basis for formulating the primary research question driving this project by incorporating the findings and knowledge derived from various studies:

To what extent is the need for correlated aircraft spare parts predictable when considering previous demand observations from available transaction records, technical aircraft documentation, and customer characteristics?

To answer this research question in a structured manner, it has been decomposed into 3 sub-questions with several underlying sub-questions:

1. How much do prior spare part demand data reflect the theoretical/technical association between aircraft spare parts for a maintenance event described in Maintenance Planning documents (MPD) and the Aircraft Maintenance Manual (AMM)?
   (a) What is the difference in correlation between conditional and unconditional identified aircraft spare parts of a specific maintenance event, as indicated in the technical maintenance documentation, based on previous demand observations?
   (b) What is the difference in the observed correlation between aircraft spare parts of a maintenance event with OEM proprietary parts and common spare parts?
   (c) What is the ideal and reasonable timeline/range for grouping purchases together to analyze the correlation between aircraft spare parts in sales records?
2. To what extent are different aircraft customer types reflected by their purchase behavior in previous demand observations?
   (a) What are the relevant subjects that describe customer purchase behavior?
   (b) How can these features be translated into variables related to sales records?
   (c) What is the optimal number of clusters to describe the similarities and discrepancies between customers?
   (d) What is the effect of customer clustering on the magnitude of the earlier examined technical correlation between spare parts?
   (e) To what extent can the statistically obtained customer cluster be identified or labeled as the known aircraft customer types by industry experts?
3. Which features positively affect demand forecasts for aircraft spare parts and how can they be incorporated into a prediction model?
   (a) Which variables are relevant to forecast demand for aircraft spare parts and are these features present in the available data resources?
   (b) What is the effect of the time horizon on the accuracy of the forecast?
   (c) What is the effect of clustering customers on the accuracy of the forecast?

Based on these research questions, a preliminary outline of the remaining part of this project is structured as follow: Firstly, the correlation between spare parts involved in similar maintenance events will be analyzed using an association rule pattern mining technique. The results will provide valuable information and additional input features for the later-developed prediction model. Next, an in-depth investigation into spare part characteristics and customer purchasing behavior will be conducted. This analysis aims to extract essential features that will enhance the prediction model's ability to forecast upcoming spare part orders based on the current customer spare part order. To complete the project, the developed prediction model will be evaluated on its performance accuracy and its practical applicability within existing company processes.

III

Supporting work

A

Overview of Spare Parts used to assess SPSO-CM

An overview of all spare parts (PNs) used to simulate and assess the performance of the proposed Subsequent PN Purchase Order Occurrence Classification Model (SPSO-CM) across two chosen MPD events (A and B), along with their specific properties, is presented in Table A.1. Additional details on variable computation and the overall assessment of the SPSO-CM can be found in Part I.

Table A.1: Overview of the PN characteristics for clustering analysis and SPSO-CM performance assessment.

TDF Information							***Spare Part Catalog***	***Sales Records***					***Cluster***
MPD	PN	Mat_{PN}	AMM Level	Conditional L1	Conditional L2	ESS	$N_{PN-inter}$	N_{PN}	$(N_{PN,customers}$	$\overline{Qnt}_{PN}$	ADI_{PN}	CV^2_{PN}	C_{PN}
5*A	A.1	70	1	-	-	3	2	263	90	3.35	1.74	3.02	C.1
	A.2	70	1	-	-	3	2	122	53	1.80	1.77	1.87	C.1
	A.3	70	1	-	-	3	4	92	51	16.59	2.80	6.64	C.1
	A.4	30	2	False	-	3	3	63	34	14.52	3.10	4.50	C.1
	A.5	10	3	False	True	2	1	432	134	8.52	1.02	1.16	C.0
23*B	B.1	30	2,3	True	True	2	1	75	39	302.97	2.26	5.78	C.2
	B.2	30	2,3	True	True	3	1	87	42	256.77	1.51	3.55	C.2
	B.3	30	2,3	True/False	True/False	2	1	99	53	255.29	2.43	3.45	C.2
	B.4	30	2	True	-	2	1	103	27	293.50	1.83	3.04	C.2
	B.5	10	2	False	-	1	7	2561	207	37.44	1.00	0.33	C.0
	B.6	10	2	False	-	1	4	996	139	5.82	1.00	0.71	C.0
	B.7	10	2	False	-	1	10	1032	132	6.68	1.00	0.60	C.0
	B.8	10	2	False	-	1	3	1316	170	6.91	1.01	0.63	C.0
	B.9	10	2	False	-	1	8	1165	166	7.52	1.00	0.61	C.0
	B.10	10	3	False	True	2	1	432	134	8.52	1.02	1.16	C.0
	B.11	70	3	True	True	2	1	69	36	7.09	2.66	4.72	C.1
	B.12	30	3	True	True	3	7	52	20	370.44	3.56	16.30	C.2
	B.13	34	3	True	True	1	1	263	116	62.29	1.27	1.94	C.1
	B.14	70	3	True	True	3	2	16	13	6.25	7.73	8.84	C.3
	B.15	30	3	True	True	2	4	115	61	518.84	2.13	9.94	C.2
	B.16	30	3	True	True	2	1	57	33	359.68	2.37	2.99	C.2
	B.17	30	3	True	True	2	1	45	30	426.64	4.60	9.43	C.2
	B.18	30	3	False	False	2	2	170	56	567.21	1.72	7.49	C.2
	B.19	30	3	False	True	3	2	24	11	15.17	5.87	13.53	C.3
	B.20	30	3	False	True	2	12	9	7	7.78	9.75	10.27	C.3
	B.21	30	3	False	True	2	12	7	7	1.86	11.25	11.23	C.3
	B.22	34	3	False	True	3	1	76	34	23.28	2.21	4.34	C.1
	B.23	34	3	False	True	3	1	26	17	3.73	5.88	10.57	C.3

B

SPSO-CM Configuration Results

This appendix provides a comprehensive summary of the computational results obtained using SPSO-CM for both MPD events selected for the scientific paper presented in Part I. Specific configuration details for MPD A and MPD B are available in Table B.1 and Table B.2, respectively. Furthermore, Table B.3 presents the prediction accuracy for each PN.

Table B.1: MPD A Results of all possible SPSO-CM configurations for a 30-day, 60-day, and 90-day time window

	30 days				**60 days**				**90 days**			
	Weighted Average				Weighted Average				Weighted Average			
Model	**Precision**	**Recall**	**F1**	**Runtime [s]**	**Precision**	**Recall**	**F1**	**Runtime [s]**	**Precision**	**Recall**	**F1**	**Runtime [s]**
$L-N-CO$	0.220	0.359	0.255	340.49	0.360	0.349	0.325	410.58	0.336	0.54	0.388	450.54
$L-C-CO$	0.265	0.308	0.257	348.76	0.362	0.556	0.408	385.67	0.337	0.526	0.400	409.41
$X-N-CO$	0.464	0.231	0.194	321.42	0.328	0.508	0.389	329.39	0.289	0.474	0.348	324.00
$X-C-CO$	0.444	0.410	0.228	320.46	0.404	0.381	0.369	312.03	0.286	0.540	0.365	313.12
$L-C-SO$	0.309	0.222	0.244	391.11	0.689	0.698	0.660	395.69	0.845	0.453	0.543	389.03
$L-N-SO$	0.546	0.148	0.167	368.22	0.699	0.674	0.631	416.28	0.647	0.622	0.525	395.89
$X-N-SO$	0.069	0.222	0.102	319.58	0.719	0.861	0.757	324.10	0.867	0.793	0.825	317.05
$X-C-SO$	0.099	0.259	0.135	321.59	0.532	0.744	0.607	326.29	0.642	0.901	0.690	327.31

Table B.2: MPD B Results of all possible SPSO-CM configurations for a 30-day, 60-day, and 90-day time window

	30 days				**60 days**				**90 days**			
	Weighted Average				Weighted Average				Weighted Average			
Model	**Precision**	**Recall**	**F1**	**Runtime [s]**	**Precision**	**Recall**	**F1**	**Runtime [s]**	**Precision**	**Recall**	**F1**	**Runtime [s]**
$L-N-CO$	0.369	0.3238	0.324	3614.30	0.484	0.482	0.469	2915.44	0.546	0.551	0.536	3506.23
$L-C-CO$	0.361	0.333	0.328	2662.58	0.484	0.487	0.471	2722.56	0.548	0.518	0.525	10462.61
$X-N-CO$	0.395	0.371	0.362	2238.23	0.476	0.446	0.448	2262.56	0.537	0.503	0.511	1544.32
$X-C-CO$	0.380	0.312	0.319	2188.67	0.475	0.457	0.451	2054.42	0.528	0.469	0.490	1918.12
$L-C-SO$	0.355	0.399	0.353	2075.60	0.418	0.462	0.424	2558.24	0.531	0.531	0.518	2670.85
$L-N-SO$	0.353	0.328	0.324	2005.89	0.437	0.496	0.453	2536.88	0.539	0.539	0.522	1826.41
$X-N-SO$	0.348	0.302	0.306	1734.23	0.452	0.493	0.488	1740.23	0.518	0.567	0.524	1620.69
$X-C-SO$	0.339	0.305	0.292	1560.78	0.462	0.496	0.498	1590.80	0.547	0.576	0.545	1709.15

Table B.3: Computational results with the most optimal configurations: X-N-SO for MPD A and X-C-SO for MPD B.

PN	TN	FP	FN	TP	Precision	Recall	F1	AUC_{PR}	MCC	ME	ME class
A.1	22	2	5	16	0.889	0.762	0.821	0.912	0.691	0.856	Good Accuracy
A.2	32	3	1	9	0.750	0.900	0.818	0.571	0.766	0.789	Good Accuracy
A.3	36	0	1	8	1.000	0.889	0.941	0.913	0.930	0.946	Good Accuracy
A.5	32	1	3	9	0.900	0.750	0.818	0.920	0.766	0.876	Good Accuracy
B.1	341	28	12	0	0.000	0.000	0.000	0.060	-0.051	0.252	Poor Accuracy
B.2	342	36	0	3	0.077	1.000	0.143	0.105	0.264	0.378	Poor Accuracy
B.3	293	12	54	22	0.647	0.289	0.400	0.385	0.351	0.534	Random Chance
B.4	317	45	6	13	0.224	0.684	0.338	0.307	0.339	0.496	Random Chance
B.5	137	38	74	132	0.776	0.641	0.702	0.801	0.425	0.732	Good Accuracy
B.6	158	54	74	95	0.638	0.562	0.597	0.660	0.313	0.643	Moderate Accuracy
B.7	181	73	71	56	0.434	0.441	0.438	0.438	0.153	0.507	Random Chance
B.8	108	106	41	126	0.543	0.754	0.632	0.623	0.264	0.629	Moderate Accuracy
B.9	108	76	56	141	0.650	0.716	0.681	0.771	0.305	0.689	Moderate Accuracy
B.10	250	53	50	28	0.346	0.359	0.352	0.407	0.182	0.485	Random Chance
B.11	312	24	11	34	0.586	0.756	0.660	0.648	0.615	0.731	Good Accuracy
B.12	345	7	18	11	0.611	0.379	0.468	0.605	0.449	0.631	Moderate Accuracy
B.13	288	34	20	39	0.534	0.661	0.591	0.650	0.511	0.688	Moderate Accuracy
B.14	380	0	1	0	0.000	0.000	0.000	0.501	0.000	0.375	Poor Accuracy
B.15	317	37	22	5	0.119	0.185	0.145	0.167	0.066	0.345	Poor Accuracy
B.16	346	1	34	0	0.000	0.000	0.000	0.137	-0.016	0.280	Poor Accuracy
B.17	304	38	21	18	0.321	0.462	0.379	0.236	0.300	0.479	Random Chance
B.18	325	26	25	5	0.161	0.167	0.164	0.100	0.091	0.339	Poor Accuracy
B.19	377	0	4	0	0.000	0.000	0.000	0.005	0.000	0.251	Poor Accuracy
B.20	138	241	0	2	0.008	1.000	0.016	0.012	0.055	0.271	Poor Accuracy
B.22	317	24	2	38	0.613	0.950	0.745	0.589	0.730	0.766	Good Accuracy
B.23	293	45	17	26	0.366	0.605	0.456	0.511	0.383	0.588	Random Chance

Bibliography

[1] Aviation Maintenance Degrees Fuel Career Opportunities, 2019.

[2] Harry A. Kinnison and Tariq Siddiqui. Aviation Maintenance Management, volume 2. Second edition, 2013. ISBN 978-0-07-180502-5.

[3] Thijs Oudkerk. Predictive Maintenance for Aircraft Systems: Using textual elements as covariates. 2019.

[4] Tiedo Tinga. Application of physical failure models to enable usage and load based maintenance. Reliability Engineering & System Safety, 95(10):1061–1075, 10 2010. ISSN 0951-8320. doi: 10.1016/J.RESS.2010.04.015.

[5] Interchangeability. URL https://www.satair.com/help-and-support/products-ordering/interchangeability.

[6] Jan Gottemeier. An Empirical Comparison of Artificial Neural Network for Forecasting Commercial Aircraft Spare Parts Demand. PhD thesis, SRH University Heidelberg, 10 2019.

[7] Darli Rodrigues Vieira and Paula Lavorato Loures. Maintenance, Repair and Overhaul (MRO) Fundamentals and Strategies: An Aeronautical Industry Overview Holds the research chair in Management of Aeronautical Projects Université du Québec à Trois Rivières-Canada. International Journal of Computer Applications, 135(12):21–29, 2016.

[8] Hamad Al-Kaabi, Andrew Potter, and Mohamed Naim. REVIEWS AND CASE STUDIES An outsourcing decision model for airlines' MRO activities. Quality in Maintenance Engineering, 13(3):217–227, 2007. doi: 10.1108/13552510710780258. URL www.emeraldinsight.com/1355-2511.htm.

[9] A. A. Syntetos, J. E. Boylan, and J. D. Croston. On the categorization of demand patterns. Journal of the Operational Research Society, 56(5):495–503, 8 2005. ISSN 01605682. doi: 10.1057/PALGRAVE.JORS.2601841/TABLES/2. URL https://link.springer.com/article/10.1057/palgrave.jors.2601841.

[10] Çerağ Pinçe, Laura Turrini, and Joern Meissner. Intermittent demand forecasting for spare parts: A Critical review. Omega (United Kingdom), 105, 12 2021. ISSN 03050483. doi: 10.1016/J.OMEGA.2021.102513.

[11] Y. S. Park and S. Lek. Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. In Developments in Environmental Modelling, volume 28, pages 123–140. Elsevier B.V., 2016. doi: 10.1016/B978-0-444-63623-2.00007-4.

[12] Greg Van Houdt, Carlos Mosquera, and Gonzalo Nápoles. A review on the long short-term memory model. Artificial Intelligence Review, 53(8):5929–5955, 12 2020. ISSN 15737462. doi: 10.1007/S10462-020-09838-1/TABLES/1.

[13] Matteo Kalchschmidt, Roberto Verganti, and Giulio Zotteri. Forecasting demand from heterogeneous customers. International Journal of Operations & Production Management, 26(6):619–638, 2006. doi: 10.1108/01443570610666975.

[14] Mahya Seyedan, Fereshteh Mafakheri, and Chun Wang. Cluster-based demand forecasting using Bayesian model averaging: An ensemble learning approach. Decision Analytics Journal, 3(100033), 6 2022. ISSN 2772-6622. doi: 10.1016/J.DAJOUR.2022.100033.

[15] Philippe Fournier-Viger, Jerry Chun, Wei Lin, Rage Uday Kiran, Yun Sing Koh, and Rincy Thomas. A Survey of Sequential Pattern Mining. 1(1), 2017.

[16] Qichen Deng, Bruno F. Santos, and Wim J. C. Verhagen. A novel decision support system for optimizing aircraft maintenance check schedule and task allocation. Decision Support Systems, 146(113545), 7 2021. ISSN 0167-9236. doi: 10.1016/J.DSS.2021.113545.

[17] Mr Shannon and P Ackert. Basics of Aircraft Maintenance Programs for Financiers: Evaluation & Insights of Commercial Aircraft Maintenance Programs. 10 2010. URL http://aircraftmonitor.com/uploads/1/5/9/9/15993320/basics_of_aircraft_maintenance_programs_for_financiers___v1.pdf.

[18] A Recall on the Correct Use of the MEL. Technical report, Airbus S.A.S., 1 2018. URL https://mms-safetyfirst.s3.eu-west-3.amazonaws.com/pdf/safety+first/a-recall-on-the-correct-use-of-the-mel.pdf.

[19] Sianez Gonzalez Velez Gallardo and Ana Yolanda Miguel Arturo. Development of an enhanced forecasting methodology for unscheduled spare part consumption in aircraft maintenance events. PhD thesis, Hamburg University of Technology, Hamburg, 10 2010.

[20] Adel A. Ghobbar and Chris H. Friend. Evaluation of forecasting methods for intermittent parts demand in the field of aviation: A predictive model. Computers and Operations Research, 30(14):2097–2114, 2003. ISSN 03050548. doi: 10.1016/S0305-0548(02)00125-9.

[21] Divish Rengasamy, Hervé P. Morvan, and Grazziela P. Figueredo. Deep Learning Approaches to Aircraft Maintenance, Repair and Overhaul: A Review. In International Conference on Intelligent Transportation Systems (ITSC), pages 150–156, Maui, Hawaii, USA, 11 2018. IEEE. URL https://ieeexplore-ieee-org.tudelft.idm.oclc.org/stamp/stamp.jsp?tp=&arnumber=8569502&tag=1.

[22] Stan Shparberg and Bob Lange. Airbus Global Market Forecast 2022-2041 Presentation, 2022.

[23] AIR Transport Action Group. Commitment To Fly Net Zero 2050, 10 2021. URL https://aviationbenefits.org/media/167501/atag-net-zero-2050-declaration.pdf.

[24] Geraldine Cros. Airline Maintenance Cost Executive Commentary. Technical report, 2 2020. URL https://www.iata.org/contentassets/bf8ca67c8bcd4358b3d004b0d6d0916f/fy2020-mctg-report_public.pdf.

[25] Lolke van der Heide. Airbus kan de productie niet aan, 11 2022. URL https://www.luchtvaartnieuws.nl/nieuws/categorie/18/technologie/airbus-kan-de-productie-niet-aan.

[26] A. A. Ghobbar and C. H. Friend. Sources of intermittent demand for aircraft spare parts within airline operations. Journal of Air Transport Management, 8(4):221–231, 2002. ISSN 09696997. doi: 10.1016/S0969-6997(01)00054-0.

[27] Jingyao Gu, Guoqing Zhang, and Kevin W. Li. Efficient aircraft spare parts inventory management under demand uncertainty. Journal of Air Transport Management, 42:101–109, 1 2015. ISSN 0969-6997. doi: 10.1016/J.JAIRTRAMAN.2014.09.006.

[28] MBA Aviation. Utilizing Alternative Collateral: Spare Parts. URL https://www.mba.aero/utilizing-alternative-collateral-spare-parts/#:~:text=Rotable%20items%20are%20parts%20that,gear%2C%20and%20major%20engine%20accessories.

[29] N. Ahmadi Mobarakeh, M. K. Shahzad, A. Baboli, and R. Tonadre. Improved Forecasts for uncertain and unpredictable Spare Parts Demand in Business Aircraft's with Bootstrap Method. 50(1):15241–15246, 7 2017. ISSN 24058963. doi: 10.1016/J.IFACOL.2017.08.2379.

[30] Andrea Bacchetti and Nicola Saccani. Spare parts classification and demand forecasting for stock control: Investigating the gap between research and practice. Omega, 40(6):722–737, 12 2012. ISSN 0305-0483. doi: 10.1016/J.OMEGA.2011.06.008.

[31] T Rad, N Shanmugarajan, and M I M Wahab. Classification of Critical Spares for Aircraft Maintenance. ICSSSM11, pages 1–6, 2011. doi: 10.1109/ICSSSM.2011.5959470.

[32] Benito E. Flores and D.Clay Whybark. Implementing multiple criteria ABC analysis. Journal of Operations Management, 7(1-2):79–85, 10 1987. ISSN 02726963. doi: 10.1016/0272-6963(87)90008-8.

[33] James AG Krupp. Are ABC codes an obsolete Technology? APICS - The performance Advantage, 4(4): 34–35, 1994.

[34] T. M. Williams. Stock Control with Sporadic and Slow-Moving Demand. Journal of the Operational Research Society 1984, 35(10):939–948, 10 1984. ISSN 1476-9360. doi: 10.1057/JORS.1984.185. URL https://link.springer.com/article/10.1057/jors.1984.185.

[35] J. E. Boylan, A. A. Syntetos, and G. C. Karakostas. Classification for forecasting and stock control: A case study. Journal of the Operational Research Society, 59(4):473–481, 10 2008. ISSN 14769360. doi: 10.1057/PALGRAVE.JORS.2602312/FIGURES/6. URL https://link.springer.com/article/10.1057/palgrave.jors.2602312.

[36] Robert G. Brown. Exponential Smoothing for Predicted Demand, 11 1956.

[37] Everette S. Gardner. Exponential smoothing: The state of the art—Part II. International Journal of Forecasting, 22(4):637–666, 10 2006. ISSN 01692070. doi: 10.1016/j.ijforecast.2006.03.005. URL https://linkinghub.elsevier.com/retrieve/pii/S0169207006000392.

[38] Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. International Journal of Forecasting, 20(1):5–10, 1 2004. ISSN 0169-2070. doi: 10.1016/J.IJFORECAST.2003.09.015.

[39] Somnath Mukhopadhyay, Adriano O. Solis, and Rafael S. Gutierrez. The accuracy of non-traditional versus traditional methods of forecasting lumpy demand. Journal of Forecasting, 31(8):721–735, 2012. ISSN 1099131X. doi: 10.1002/FOR.1242.

[40] J. D. Croston. Forecasting and Stock Control for Intermittent Demands. Journal of the Operational Research Society 1972 23:3, 23(3):289–303, 9 1972. ISSN 1476-9360. doi: 10.1057/JORS.1972.50.

[41] Aris A. Syntetos and John E. Boylan. The accuracy of intermittent demand estimates. International Journal of Forecasting, 21(2):303–314, 4 2005. ISSN 01692070. doi: 10.1016/j.ijforecast.2004.10.001.

[42] Thomas R. Willemain, Charles N. Smart, Joseph H. Shockor, and Philip A. DeSautels. Forecasting intermittent demand in manufacturing: a comparative evaluation of Croston's method. International Journal of Forecasting, 10(4):529–538, 12 1994. ISSN 01692070. doi: 10.1016/0169-2070(94)90021-3.

[43] A. H.C. Eaves and B. G. Kingsman. Forecasting for the ordering and stock-holding of spare parts. https://doi.org/10.1057/palgrave.jors.2601697, 55(4):431–437, 2017. ISSN 14769360. doi: 10.1057/PALGRAVE.JORS.2601697.

[44] Ruud H. Teunter, Aris A. Syntetos, and M. Zied Babai. Intermittent demand: Linking forecasting to inventory obsolescence. European Journal of Operational Research, 214(3):606–615, 11 2011. ISSN 0377-2217. doi: 10.1016/J.EJOR.2011.05.018.

[45] Raghad Hemeimat, Lina Al-Qatawneh, Mazen Arafeh, and Shadi Masoud. Forecasting Spare Parts Demand Using Statistical Analysis. American Journal of Operations Research, 06(02), 3 2016. ISSN 2160-8830. doi: 10.4236/ajor.2016.62014.

[46] George E. P. Box, Gwilym M Jenkins, Gregory C. Reinsel, and Greta M. Ljung. Time Series Analysis : Forecasting and Control. John Wiley & Sons, Incorporated, Hoboken, 5 edition, 2016. ISBN 9781118674918.

[47] Sima Siami-Namini, Neda Tavakoli, and Akbar Siami Namin. A Comparison of ARIMA and LSTM in Forecasting Time Series. Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, pages 1394–1401, 1 2019. doi: 10.1109/ICMLA.2018.00227.

[48] Anjali Gautam and Vrijendra Singh. Parametric Versus Non-Parametric Time Series Forecasting Methods: A Review. Journal of Engineering Science and Technology Review, 13(3):165–171, 2020. doi: 10.25103/jestr.133.18. URL www.jestr.org.

[49] Vishant Kumar, Ramesh Kumar, and R K Garg. THE DEMAND FORECASTING: A COMPARATIVE REVIEW OF CONVENTIONAL AND NON-CONVENTIONAL TECHNIQUES. www.tjprc.org SCOPUS Indexed Journal editor@tjprc.org, 2019. URL www.tjprc.org.

[50] Sean J. Taylor and Benjamin Letham. Forecasting at Scale. https://doi.org/10.1080/00031305.2017.1380080, 72(1):37–45, 1 2018. ISSN 15372731. doi: 10.1080/00031305.2017.1380080. URL https://www.tandfonline.com/doi/abs/10.1080/00031305.2017.1380080.

[51] Martim Sousa, Ana Maria Tomé, and José Moreira. Long-term forecasting of hourly retail customer flow on intermittent time series with multiple seasonality. Data Science and Management, 5(3):137–148, 9 2022. ISSN 2666-7649. doi: 10.1016/J.DSM.2022.07.002.

[52] Tim Hill, Marcus O'Connor, and William Remus. Neural Network Models for Time Series Forecasts. https://doi.org/10.1287/mnsc.42.7.1082, 42(7):1082–1092, 7 1996. ISSN 00251909. doi: 10.1287/MNSC.42.7.1082. URL https://pubsonline.informs.org/doi/abs/10.1287/mnsc.42.7.1082.

[53] Ronald L Iman and William Jay Conover. Modern business statistics. John Wiley & Sons, Inc., 1988.

[54] Rafael S. Gutierrez, Adriano O. Solis, and Somnath Mukhopadhyay. Lumpy demand forecasting using neural networks. International Journal of Production Economics, 111(2):409–420, 2 2008. ISSN 0925-5273. doi: 10.1016/J.IJPE.2007.01.007.

[55] Daniel Graupe. Principles Of Artificial Neural Networks. 3 edition, 2013. ISBN 9789814522748.

[56] Merve Sahin, Merve Şahin, Recep Kızılaslan, and Ömer F Demirel. Forecasting Aviation Spare Parts Demand Using Croston Based Methods and Artificial Neural Networks. Journal of Economic and Social Research, 15(2):1–21, 2013.

[57] Alexis Wieland and Russell Leighton. Geometric analysis of neural network capabilities. San Diego, CA, 1987.

[58] RP Rippmann. An introduction to computing with Neural Networks. IEEE ASSP Mag, 1(110.2), 1987.

[59] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, 65(6):386–408, 1958. ISSN 1939-1471. doi: 10.1037/h0042519.

[60] Siddharth Sharma, Simone Sharma, and Anidhya Athaiya. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. International Journal of Engineering Applied Sciences and Technology, 4:310–316, 2020. URL http://www.ijeast.com.

[61] Robby Henkelmann. A Deep Learning based Approach for Automotive Spare Part Demand Forecasting. PhD thesis, Otto-von-Guericke_University, Magdeburg, 2018. URL https://www.is.ovgu.de/is_media/Master+und+Bachelor_Arbeiten/MasterThesis_RobbyHenkelmann-download-1-p-4746.pdf.

[62] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. Nature 1986 323:6088, 323(6088):533–536, 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL https://www.nature.com/articles/323533a0.

[63] Backpropagation in data mining, 2022. URL https://www.geeksforgeeks.org/backpropagation-in-data-mining/.

[64] Perceptron, 4 2020. URL https://nl.wikipedia.org/wiki/Perceptron.

[65] M. Z. Babai, A. Tsadiras, and C. Papadopoulos. On the empirical performance of some new neural network methods for forecasting intermittent demand. IMA Journal of Management Mathematics, 31:281–305, 2020. doi: 10.1093/imaman/dpaa003.

[66] Zhengyu Ma, Chen Wang, and Zhongze Zhang. Deep Learning Algorithms for Automotive Spare Parts Demand Forecasting. Proceedings - 2021 International Conference on Computer Information Science and Artificial Intelligence, CISAI 2021, pages 358–361, 2021. doi: 10.1109/CISAI54367.2021.00075.

[67] C. A. Mitrea, C. K.M. Lee, and Z. Wu. A comparison between neural networks and traditional forecasting methods: A case study. International Journal of Engineering Business Management, 1(2):19–24, 9 2009. ISSN 18479790. doi: 10.5772/6777/FORMAT/EPUB.

[68] Recurrent Neural Networks (RNN) - The Vanishing Gradient Problem, 2018. URL https://www.superdatascience.com/blogs/recurrent-neural-networks-rnn-the-vanishing-gradient-problem.

[69] Josef Hochreiter. Untersuchungen zu dynamischen neuronalen Netzen. PhD thesis, Technische Universitat Munchen, Munchen, 6 1991.

[70] Yoshua Bengio, Paolo Frasconi, and Patrice Simard. Problem of learning long-term dependencies in recurrent networks. 1993 IEEE International Conference on Neural Networks, pages 1183–1188, 1993. doi: 10.1109/ICNN.1993.298725.

[71] M. R. Amin-Naseri and B. Rostami Tabar. Neural network approach to lumpy demand forecasting for spare parts in process industries. Proceedings of the International Conference on Computer and Communication Engineering 2008, ICCCE08: Global Links for Human Development, pages 1378–1382, 2008. doi: 10.1109/ICCCE.2008.4580831.

[72] F. Lolli, R. Gamberini, A. Regattieri, E. Balugani, T. Gatos, and S. Gucci. Single-hidden layer neural networks for forecasting intermittent demand. International Journal of Production Economics, 183: 116–128, 1 2017. ISSN 0925-5273. doi: 10.1016/J.IJPE.2016.10.021.

[73] Badar ul Islam and Shams Forruque Ahmed. Short-Term Electrical Load Demand Forecasting Based on LSTM and RNN Deep Neural Networks. Mathematical Problems in Engineering, 2022:1–10, 7 2022. ISSN 1563-5147. doi: 10.1155/2022/2316474.

[74] Wei Wang, Jiayu Chen, Tianzhen Hong, and Na Zhu. Occupancy Prediction through Markov based Feedback Recurrent Neural Network (M-FRNN) Algorithm with WiFi Probe Technology. Building and Environment, 138:160–170, 6 2018. ISSN 03601323. doi: 10.1016/j.buildenv.2018.04.034.

[75] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, 11 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735.

[76] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. Neural Computation, 12(10):2451–2471, 2000. ISSN 08997667. doi: 10.1162/089976600300015015.

[77] Kiran Kumar Chandriah and Raghavendra V. Naraganahalli. RNN / LSTM with modified Adam optimizer in deep learning approach for automobile spare parts demand forecasting. Multimedia Tools and Applications, 80(17):26145–26159, 7 2021. ISSN 15737721. doi: 10.1007/S11042-021-10913-0/FIGURES/5.

[78] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, pages 1724–1734, 6 2014. doi: 10.48550/arxiv.1406.1078.

[79] Rui Fu, Zuo Zhang, and Li Li. Using LSTM and GRU neural network methods for traffic flow prediction. Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016, pages 324–328, 1 2017. doi: 10.1109/YAC.2016.7804912.

[80] Corinna Cortes, Vladimir Vapnik, and Lorenza Saitta. Support-vector networks. Machine Learning 1995 20:3, 20(3):273–297, 9 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. URL https://link.springer.com/article/10.1007/BF00994018.

[81] Vladimir Vapnik, Steven E Golowich, and Alex Smola. Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing·. Proceedings of the 9th International Conference on Neural Information Processing Systems, pages 281–287, 12 1996.

[82] Abdeslem Boukhtouta and Peter Jentsch. Support vector machine for demand forecasting of canadian armed forces spare parts. Proceedings - 6th International Symposium on Computational and Business Intelligence, ISCBI 2018, pages 59–64, 2 2019. doi: 10.1109/ISCBI.2018.00021.

[83] L. J. Cao and Francis E.H. Tay. Support vector machine with adaptive parameters in financial time series forecasting. IEEE Transactions on Neural Networks, 14(6):1506–1518, 11 2003. ISSN 10459227. doi: 10.1109/TNN.2003.820556.

[84] Yukun Bao, Wen Wang, and Jinlong Zhang. Forecasting intermittent demand by SVMs regression. Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics, 1:461–466, 2004. ISSN 1062922X. doi: 10.1109/ICSMC.2004.1398341.

[85] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. Advances in Neural Information Processing Systems, 2017-December:5999–6009, 6 2017. ISSN 10495258. doi: 10.48550/arxiv.1706.03762. URL https://arxiv.org/abs/1706.03762v5.

[86] Neo Wu, Bradley Green, Xue Ben, and Shawn O'Banion. Deep Transformer Models for Time Series Forecasting: The Influenza Prevalence Case. 1 2020. doi: 10.48550/arxiv.2001.08317. URL https://arxiv.org/abs/2001.08317v1.

[87] Thomas R. Willemain, Charles N. Smart, and Henry F. Schwarz. A new approach to forecasting intermittent demand for service parts inventories. International Journal of Forecasting, 20(3):375–387, 7 2004. ISSN 0169-2070. doi: 10.1016/S0169-2070(03)00013-X.

[88] Aris A. Syntetos, M. Zied Babai, and Everette S. Gardner. Forecasting intermittent inventory demands: simple parametric methods vs. bootstrapping. Journal of Business Research, 68(8):1746–1752, 8 2015. ISSN 01482963. doi: 10.1016/j.jbusres.2015.03.034.

[89] Nadine Côrte-Real, Pedro Ruivo, Tiago Oliveira, and Aleš Popovič. Unlocking the drivers of big data analytics value in firms. Journal of Business Research, 97:160–173, 4 2019. ISSN 0148-2963. doi: 10.1016/J.JBUSRES.2018.12.072.

[90] Shu Ching Wang, Yao Te Tsai, and Yi Syuan Ciou. A hybrid big data analytical approach for analyzing customer patterns through an integrated supply chain network. Journal of Industrial Information Integration, 20:100177, 12 2020. ISSN 2452-414X. doi: 10.1016/J.JII.2020.100177.

[91] Paul W. Murray, Bruno Agard, and Marco A. Barajas. Forecasting Supply Chain Demand by Clustering Customers. IFAC-PapersOnLine, 48(3):1834–1839, 1 2015. ISSN 2405-8963. doi: 10.1016/J.IFACOL.2015.06.353.

[92] Federico Caniato, Matteo Kalchschmidt, Stefano Ronchi, Roberto Verganti, Giulio Zotteri, Federico Caniatoy, Matteo Kalchschmidtz, Stefano Ronchiy, and Roberto Vergantiy. Clustering customers to forecast demand, Production Planning & Control. Production Planning & Control, 16(1):32–43, 2005. ISSN 1366-5871. doi: 10.1080/09537280512331325155.

[93] Arthur Middleton Hughes. Strategic Database Marketing. McGraw-Hill, New York, 4 edition, 2012.

[94] Hülya Güçdemir and Hasan Selim. Integrating multi-criteria decision making and clustering for business customer segmentation. Industrial Management & Data Systems, 115(6):1022–1040, 2015. doi: 10.1108/IMDS-01-2015-0027.

[95] Manish Verma, Mauly Srivastava, Neha Chack, Atul Kumar Diswar, and Nidhi Gupta. A Comparative Study of Various Clustering Algorithms in Data Mining. International Journal of Engineering Research and Applications (IJERA), 2:1379–1384, 2012. URL www.ijera.com.

[96] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos Artemio Coello Coello. A survey of multiobjective evolutionary algorithms for data mining: Part i. IEEE Transactions on Evolutionary Computation, 18(1):4–19, 2 2014. ISSN 1089778X. doi: 10.1109/TEVC.2013.2290086.

[97] Marina Santini. Advantages & Disadvantages of k-Means and Hierarchical clustering (Unsupervised Learning), 2016. URL http://santini.se/teaching/ml/2016/Lect_10/10c_UnsupervisedMethods.pdf.

[98] Girish Punj and David W. Stewart. Cluster Analysis in Marketing Research: Review and Suggestions for Application. https://doi.org/10.1177/002224378302000204, 20(2):134–148, 12 2018. ISSN 0022-2437. doi: 10.1177/002224378302000204. URL https://journals.sagepub.com/doi/full/10.1177/002224378302000204.

[99] R. J. Kuo, L. M. Ho, and C. M. Hu. Integration of self-organizing feature map and K-means algorithm for market segmentation. Computers & Operations Research, 29(11):1475–1493, 9 2002. ISSN 0305-0548. doi: 10.1016/S0305-0548(01)00043-0.

[100] Saibal Dutta, Sujoy Bhattacharya, and Kalyan Kumar Guin. Data mining in market segmentation: A literature review and suggestions. Advances in Intelligent Systems and Computing, 335:87–98, 2015. ISSN 21945357. doi: 10.1007/978-81-322-2217-0{_}8/TABLES/3. URL https://link.springer.com/chapter/10.1007/978-81-322-2217-0_8.

[101] Teuvo Kohonen. The Self-Organizing Map. Proceedings of the IEEE, 78(9):1464–1480, 1990. ISSN 15582256. doi: 10.1109/5.58325. URL https://www.vaughn.edu/blog/aviation-maintenance-degrees-fuel-career-opportunities/.

[102] Rui Xu and Donald C. Wunsch. Clustering. John Wiley & Sons, Inc., Hoboken, NJ, USA, 10 2008. ISBN 9780470382776. doi: 10.1002/9780470382776.

[103] Jih Jeng Huang, Gwo Hshiung Tzeng, and Chorng Shyong Ong. Marketing segmentation using support vector clustering. Expert Systems with Applications, 32(2):313–317, 2 2007. ISSN 0957-4174. doi: 10.1016/J.ESWA.2005.11.028.

[104] Melody Y. Kiang, Michael Y. Hu, and Dorothy M. Fisher. An extended self-organizing map network for market segmentation — a telecommunication example. Decision Support Systems, 42(1):36–47, 10 2006. ISSN 01679236. doi: 10.1016/j.dss.2004.09.012.

[105] I.B. Sanchez, I.D. Espinos, L. Moreno Sarrion, A.Q. Lopez, and I.N. Burgos. Clients segmentation according to their domestic energy consumption by the use of self-organizing maps. In 2009 6th International Conference on the European Energy Market, pages 1–6. IEEE, 5 2009. ISBN 978-1-4244-4455-7. doi: 10.1109/EEM.2009.5207172.

[106] R. J. Kuo, K. Chang, and S. Y. Chien. Integration of Self-Organizing Feature Maps and Genetic-Algorithm-Based Clustering Method for Market Segmentation. Journal of Organizational Computing and Electronic Commerce, 14(1):43–60, 2009. ISSN 10919392. doi: 10.1207/s15327744joce1401{_}3. URL https://doi.org/10.1207/s15327744joce1401_3.

[107] Ujjwal Maulik and Sanghamitra Bandyopadhyay. Genetic algorithm-based clustering technique. Pattern Recognition, 33(9):1455–1465, 9 2000. ISSN 0031-3203. doi: 10.1016/S0031-3203(99)00137-5.

[108] M. C. Cowgill, R. J. Harvey, and L. T. Watson. A genetic algorithm approach to cluster analysis. Computers & Mathematics with Applications, 37(7):99–108, 4 1999. ISSN 0898-1221. doi: 10.1016/S0898-1221(99)00090-5.

[109] Kyoung-jae Kim and Hyunchul Ahn. Using a Clustering Genetic Algorithm to Support Customer Segmentation for Personalized Recommender Systems. In 13th International Conference on AI, Simulation, and Planning in High Autonomy Systems, pages 409–415, Seoul, 2004. URL https://link.springer.com/chapter/10.1007/978-3-540-30583-5_44.

[110] Hsiang-Hsi Liu and Chorng-Shyong Ong. Variable selection in clustering for marketing segmentation using genetic algorithms. Expert Systems with Applications, 34(1):502–510, 1 2008. ISSN 09574174. doi: 10.1016/j.eswa.2006.09.039.

[111] Anirban Mukhopadhyay, Ujjwal Maulik, Sanghamitra Bandyopadhyay, and Carlos A.Coello Coello. Survey of multiobjective evolutionary algorithms for data mining: Part II. IEEE Transactions on Evolutionary Computation, 18(1):20–35, 2014. ISSN 1089778X. doi: 10.1109/TEVC.2013.2290082.

[112] Chiheb-Eddine Ben Ncir, Mohamed Ben Mzoughia, Alaa Qaffas, and Waad Bouaguel. Evolutionary multi-objective customer segmentation approach based on descriptive and predictive behaviour of customers: application to the banking sector. Journal of Experimental & Theoretical Artificial Intelligence, pages 1–23, 5 2022. ISSN 0952-813X. doi: 10.1080/0952813X.2022.2078886.

[113] Azarnoush Ansari and Arash Riasi. Customer Clustering Using a Combination of Fuzzy C-Means and Genetic Algorithms. International Journal of Business and Management, 11(7):59–66, 6 2016. ISSN 1833-8119. doi: 10.5539/ijbm.v11n7p59.

[114] Clarisse Dhaenens and Laetitia Jourdan. Metaheuristics for data mining: survey and opportunities for big data. Annals of Operations Research, (314):117–140, 7 2022. ISSN 15729338. doi: 10.1007/S10479-021-04496-0.

[115] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In 1993 ACM SIGMOID international conference on Management of data, pages 207–216. Association for Computing Machinery (ACM), 1993. doi: 10.1145/170035.170072.

[116] Rakesh Agrawal and Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules. In Very large data bases, pages 487–499, 1994. URL https://www.vldb.org/conf/1994/P487.PDF.

[117] M Chen and H Wu. An association-based clustering approach to order batching considering customer demand patterns. Omega, 33(4):333–343, 8 2005. ISSN 03050483. doi: 10.1016/j.omega.2004.05.003.

[118] D. Magdalene Delighta Angeline. Association Rule Generation for Student Performance Analysis using Apriori Algorithm. The SIJ Transactions on Computer Science Engineering & its Applications (CSEA), 1 (1):12–16, 3 2013.

[119] Peter P. Wakabi-Waiswa and Venansius Baryamureeba. Extraction of Interesting Association Rules using Genetic Algorithms. International Journal of ICT Research, 2(1):26–33, 6 2008.

[120] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. Proceedings - International Conference on Data Engineering, pages 3–14, 1995. doi: 10.1109/ICDE.1995.380415.