Abhilash Ramanathapuram Anand

Demand Forecasting Based On Short Univariate Time Series: A Comparative Study

Master's thesis in Global Manufacturing Management
Supervisor: Fabio Sgarbossa
February 2021

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abhilash Ramanathapuram Anand

Demand Forecasting Based On Short Univariate Time Series: A Comparative Study

Master's thesis in Global Manufacturing Management
Supervisor: Fabio Sgarbossa
February 2021

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of Science and Technology

Demand Forecasting based on Short Univariate Time Series: A Comparative Study

TPK4930

Master's Thesis in Global Manufacturing Management
Department of Mechanical and Industrial Engineering

Abhilash Ramanathapuram Anand
February 2021

Academic Supervisor: Fabio Sgarbossa
Professor of Industrial Logistics
Department of Mechanical Engineering and Production

Academic Co-Supervisor: Mirco Peron
Scientist
Department of Mechanical Engineering and Production

Company Supervisor: Truls Øksnevad
Logistics engineer, Scale AQ

Preface

This report has been written in Fall 2020 to fulfil the requirements of the Master's degree program, Global Manufacturing Management, and has been submitted to the department of Mechanical and Industrial Engineering at Norwegian University of Science and Technology (NTNU). This work is done in collaboration with Scale AQ, Trondheim, Norway.

Firstly, I would like to thank Scale AQ for providing the opportunity to write my master thesis with them. Huge thanks to my company supervisor, Truls Øksnevad, for his valuable guidance and support.

Secondly, my sincere gratitude to my supervisor at NTNU, Fabio Sgarbossa, for his support and cooperation throughout the thesis. I would also like to thank my co-supervisor, Mirco Peron, for his valuable feedback and guidance in writing the report.

Finally, a big hug and thanks to my wife who has been my pillar of support for the entire master's journey. Thanks to my family for checking up on me everyday and motivating me throughout the semester. I would also like to thank all my friends, colleagues and everyone else who helped and supported me in finishing this work.

February 2021
Abhilash Ramanathapuram Anand

Summary

Demand forecasting has been studied extensively because it serves as an input to other decision processes in an organisation. Imprecise forecasts can lead to stock-outs, lost-sales or overstocking, thus not meeting the service level targets. The case company, Scale AQ, experienced similar challenges with forecasting the demand of one of their crucial components. Scale AQ is a global supplier of technology and infrastructure for land- and sea-based aquaculture. The product of focus in this study is their best-selling circular sea-based fish farming cage, cage-P. The company faced component shortage which is the brackets in the cage, especially during the peak production season, due to long replenishment lead-times involved. Consequently, reliable forecasts for atleast three-four quarters ahead were required.

To resolve the issue of limited component availability, we have chosen to improve the demand forecasts which will thereby reduce the uncertainty in demand and events of stock-outs. Since the demand of the component is dependent on the demand of the cage, we have chosen to perform demand forecasting for cage-P. Numerous demand forecasting methods, both qualitative and quantitative, have been researched since the past few decades, and the most popular and widely studied field was time-series forecasting. Thus, the main objective of this study is to investigate various time-series forecasting methods and choose a suitable method for forecasting the demand of cages.

Various time-series forecasting models were identified using systematic literature review. Then the identified models that met the selection criteria were further shortlisted for quantitative modeling. Four traditional models: Seasonal naïve, Holt-Winters(HW), State-space model (ETS) and seasonal autoregressive integrated moving average (SARIMA) and four advanced models: Prophet, Multi-layer perceptron (MLP), Long Short Term Memory (LSTM) and Support vector regressor (SVR) were selected for the comparative forecasting analysis using short univariate time series data. Each model was optimized using grid search method where optimal parameters (or hyperparameters) were selected for each model configuration. The resulting model configuration was utilized to create multi-step ahead forecasts (for four quarters ahead) and was evaluated using two performance metrics, RMSE and R^2.

The models were then compared against each other and against currently employed statistical forecasting model (seasonal naïve). It was found that most of the traditional methods outperformed the advanced methods when dealing with short univariate times-series though LSTM was found to be the best overall performing model. It was identified that the forecasting performance of all the models, except SVR, surpassed seasonal naïve model. The forecasting performance of LSTM model was found to be 51% better than seasonal naive. Whereas the SARIMA model (and its variants) resulted in an improvement of 29-48% compared to the seasonal naïve model's forecasting performance. Since traditional models performed better than advanced models on short time-series, it is recommended that Scale AQ employs SARIMA model for deriving statistical forecasts for cage-P and the other cages. It is also recommended that the statistical forecasts are complemented with managerial judgements since the domain knowledge of managers is vital in a complex environment.

This study contributes to both scientific community and the case company. With regards to scientific contribution, the study suggests an appropriate and optimized method for forecasting short univariate time-series. With regards to the case company, a suitable optimized model was recommended along with the implementation procedure to be used for the demand forecasting of cage-P as well as the other cages instead of the currently employed statistical forecasting model.

Keywords: demand forecasting, univariate time-series, traditional models, advanced models

Table of Contents

List of Tables

List of Figures

Abbreviations

MA	=	Moving Average
ES	=	Exponential Smoothing
HW	=	Holt-Winters
ETS	=	State space model with Error, Trend and Seasonality
SARIMA	=	Seasonal autoregressive integrated moving average
MLP	=	Multilayer perceptron
LSTM	=	Long Short Term Memory
SVR	=	Support Vector Regressor
RMSE	=	Root Mean Squared Error

Chapter 1

Introduction

This chapter introduces the background for this thesis and is followed by the description of the problem that advocates the research area of interest. The remaining sections briefly explain the research objective, formulation of the research questions, explanation of the research scope, and ends with the outline of the report structure.

1.1 Background

Supply chain activities involve the flow of goods from suppliers to the final customer (Chopra and Meindl 2016). Many factors affect the supply chain performance, but the most critical factor being the demand planning and forecasting activity because all the other processes are dependent on this factor (Salais-Fierro et al. 2020; Syntetos et al. 2016). Demand forecasts serve as an essential input in the decision processes of operations management because they provide information on future demand (William J. Stevenson 2014). The operations of a company in the supply chain are greatly affected by imprecise forecasts in terms of stock-outs and lost sales, or over-stocking, while not meeting service level targets (Feizabadi 2020; Syntetos et al. 2016). This reason compels the forecasters to minimize the forecast errors as much as possible. Forecast errors are measured because it gives the managers a better perspective on the deviation from the actual values which will substantiate their decision-making process (Rob J. Hyndman and Anne B. Koehler 2006). The convoluted nature of most real-world variables due to their random variation makes it complicated to accurately predict the future values of those variables regularly (Nahmias and Olsen 2020; William J. Stevenson 2014).

Demand forecasting approaches could be broadly classified as qualitative and quantitative approaches. Qualitative forecasting methods rely on human judgment and are generally subjective in nature. They are used when historical data is unavailable. Qualitative techniques such as the market research and surveys, Delphi method, and life-cycle analogical method (Chopra and Meindl 2016; William J. Stevenson 2014) are the most popular methods. On the other hand, quantitative methods are used when there is availability of historical data. Quantitative forecasting methods can be categorized into two types, namely, time-series and causal. Causal forecasting methods involved the determination of factors or explanatory variables (such as the state of the economy, interest rate, price promotions and so on) which relate to the demand to be predicted (Chopra and Meindl 2016; William J. Stevenson 2014). Time series forecasting methods involve the projection of future values of a variable based entirely on the past and present observations of the demand (Chopra and Meindl 2016; Rob J Hyndman and Athanasopoulos 2018; William J. Stevenson 2014). Various time-series forecasting techniques have been developed and they are categorised into two distinct groups, traditional forecasting techniques and advanced forecasting techniques. Traditional forecasting methods include models which extrapolates the past time series structure into the future (Rob J Hyndman and Athanasopoulos 2018). Advanced time-series forecasting methods were established which used the machine learning algorithms such as artificial neural networks, fuzzy logic, decision trees and so on to create forecasts (Carbonneau et al. 2008; Papacharalampous et al. 2018; Salais-Fierro et al. 2020).

Traditional time-series forecasting methods have performed well to date, but they do have few drawbacks such as they do not perform well when there are multiple seasons in the historical data, non-linear trends, shifts in trend, and presence of missing data and outliers. Also, it is difficult to include the effects of additional factors (such as price promotions, holidays, etc.) in most of the traditional methods (Brownlee 2017; Rob J Hyndman and Athanasopoulos 2018). Advanced forecasting methods were developed to counter the drawbacks of the classical techniques. It combines learning algorithms to identify underlying patterns, demand drivers and uncover new insights by processing an excessive number of additional factors, and determining the ones that are significant(Bouktif et al. 2018; Carbonneau et al. 2008; Jung-Pin et al. 2020; Martínez-Álvarez et al. 2015). Advanced methods exceed traditional methods because of the availability of huge amount of historical data and access to external data. This reason leads us to the conclusion that advanced methods are data-hungry methods, but there are few research articles which demonstrated the dominance of advanced methods over traditional methods with limited data (Abbasimehr et al. 2020; Abdel-Aal 2008; Delic 2019; Ismail Fawaz et al. 2019; Yu et al. 2018). Thus, a suitable method for forecasting short univariate time-series remained as an inconclusive result.

In this study, we have investigated the performance of advanced forecasting methods on a short univariate time-series with demand as the only predictor and compared it against the performance of traditional time-series methods. The methods are also bench-marked against the current statistical forecasting method (seasonal naive) used by the company to test the dominance of complex methods. These steps enabled us to select a suitable forecasting method for demand forecasting using short univariate time-series. The demand data used for the analysis is obtained from a real-life case study company, Scale AQ.

1.2 Problem Description

Scale AQ is the case company involved in this study. The company is a global supplier of technology and infrastructure for land- and sea-based aquaculture. This thesis is written in collaboration with the sea-based section of the company, and focuses on one of their key product group; circular fish farming cages. A circular fish farming cage is constructed using numerous components, however the structure could be broken down to the following main categories: (i) Polyethylene (PE) pipes; float pipes and sinker tube (ii) Brackets (iii) Walkways (iv) Net(s) (v) Others. A schematic diagram of the cage is shown in figure 1.1. The cage components are sourced from multiple suppliers in Europe and Asia. Hence some of the key components are subjected to long shipping routes (which involve long transit time), which drastically increases the overall replenishment lead-time of the components. One of the key components are the brackets shown in figure 1.2.

Brackets function as the cage "skeleton" and are connected to each other by steel rods which enables the whole structure to distribute external forces throughout the whole cage circumference. They are also used as mooring points and to keep the float pipes at a fixed distance from each other. Generally, the replenishment lead-time is approximately three to four months, but sometimes it could be as high as five months. Owing to the long replenishment lead-time of the brackets, it was difficult to predict and meet the unanticipated surge in cage demand during the peak-production season (January-May) especially for the cages with short deliv-

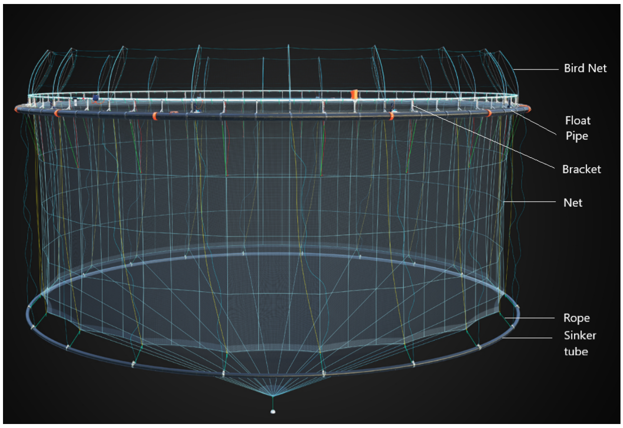

Figure 1.1: A schematic diagram of a cage

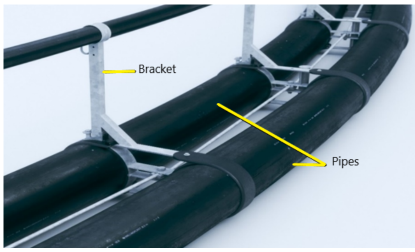

Figure 1.2: A close-up view of the bracket

ery lead-time. There are numerous ways to approach this problem of improving component availability, but we have chosen to go forward with improving the forecast which will thereby reduce the variability in demand and potential occurrences of stock-outs.

The company mostly relies on seasonal naive forecasting and judgmental forecasting as their primary forecasting methods. They required a forecasting method that would provide them with a higher forecasting accuracy, and allow lesser dependence on key personnel. Only few time-series observations were available. We have demonstrated in this thesis on how to choose a suitable forecasting method for demand forecasting using short time-series. There were four types of different brackets required for the assembly of cage-P. The bracket requirements are shown in the figure 1.3. The cage cannot be assembled even if one type of bracket is unavailable. We decided to perform forecasting on the end product instead of forecasting each type of bracket because the demand of the brackets depends on the cage demand. The final predicted demand of the cage could be dis-aggregated to component level based on the bracket requirement shown in figure 1.3.

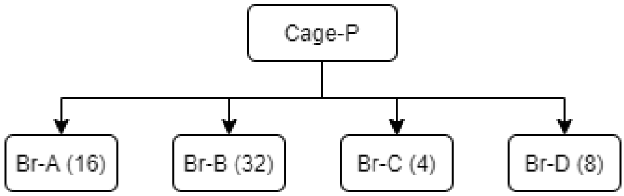

Figure 1.3: Bracket requirements for each cage

1.3 Research Objective and Research Questions

Croom (2010) explains the first step in research design as "stating the problem, system or domain in the form of research questions, propositions or constructs that define precisely what will be investigated." This section defines the research objective and the research questions that the study aims to answer.

This thesis aims at assisting the case company in choosing a suitable technique for forecasting their product demand. A variety of methods (including advanced forecasting methods and traditional forecasting methods) are chosen for model-fitting and they are all compared against each other based on the forecast performance metrics to find the best performing model.

From the background section, we know that advanced forecasting methods generally perform much better than traditional methods due to its ability of handling outliers, missing data, multiple seasonality and using unlimited data sources. We have investigated if the advanced forecasting models were able to deliver the claimed superior performance with limited data in order to find a suitable forecasting method. The research questions that will assist us in achieving this objective is given below.

- **RQ1.** ***"What are the different state-of-the-art traditional and advanced forecasting methods that can be employed on short univariate time-series data?"***

  Literature review was performed to understand the concepts of different forecasting methods and their performance metrics. Based on some selection criteria mentioned in Chapter 5, only few models were chosen for the analysis in this study. Data transformation techniques and forecast performance metrics were also investigated to aid in the evaluation of quantitative analysis.

- **RQ2.** ***"Do advanced time-series forecasting methods perform better than traditional time-series forecasting methods when dealing with short univariate time-series data?"***

  Data preprocessing was performed on the in-sample data to understand the nature of the data, the underlying patterns and components, and prepare the data for modeling. Then the various chosen models are fitted and compared against each other based on the forecast performance metrics to find the best performing model. The best performing model could be later used for creating out-of-sample forecasts.

1.4 Research Scope

This thesis only deals with the demand forecasting of one product. Though the company manufactures different cages, we have focused only on their best selling cage. The demand forecasting is done on the end product rather than on components since it is easier to perform modeling and analysis, and the demand of the components depend on the demand of the end product. The aggregated demand can then be dis-aggregated to the respective components based on the bill-of-materials (BOM) of the product. The company has primarily two types of customers, one is local and the other is global. The global or export projects' demand is known in advance and therefore is not used in our study. We have only focused on the local customers for which the demand is unknown and variable.

Time series analysis is strictly restricted to univariate time series. Multivariate time series analysis is not included in the study and no exogenous variables were used to add any additional information. This study focuses on predicting the demand in four-quarters of 2019 which is considered to be the test set. Multi-step forecasting was applied since we are predicting the demand for four quarters. The year 2020 was not included in our analysis due to COVID-19. The production was stalled for few months and was pushed to the later months. Therefore, this data cannot be used as a test set as it produces unusual seasonality and the analysis will present erroneous results leading to difficulty in choosing the best forecasting model. Also, an event such as a pandemic cannot be predicted (using a literature review or by any other means) and thus it cannot be added as a additional information to the models. These reasons compelled us to neglect the year 2020 in the analysis and use the year 2019 as the test set.

1.5 Report Structure

Chapter 1 - Introduction
Chapter 1 introduces the thesis through background and problem description. It also presents the research objective, research questions, research scope and the report structure.

Chapter 2 - Research Methodology
Chapter 2 describes the research methodology followed in this thesis.

Chapter 3 - Case study: Scale AQ
Chapter 3 presents a brief description of the case-company involved, their supply chain and their current forecasting practice are discussed.

Chapter 4 - Forecasting Models
Chapter 4 provides the forecasting models that will be used for the quantitative analysis in the later chapter and is divided into many relevant sub-chapters. It discusses time-series forecasting, the various forecasting methods, qualitative comparison of the models, selection of models, and finally, detailed mathematical model descriptions.

Chapter 5 - Model Evaluation
Chapter 5 is an integral chapter. It begins with data description and preprocessing methods where we understand the underlying components in the data. Data transformation techniques, grid search, walk-forward validation and performance metrics are discussed. Then we proceed to model implementation and configuration where the chosen traditional and advanced models are fit on the training data, and finally, the results of each model are evaluated on the test data.

Chapter 6 - Discussions
Chapter 6 presents the findings of the research, answers the research questions and discusses the findings to highlight research gaps.

Chapter 7 - Conclusion
Chapter 7 summarizes the entire work done followed by contribution, limitations and challenges, and concludes the thesis by presenting the future scope of the work.

Chapter 2

Research Methodology

Methodology is a theoretical and structured analysis of applicable methods in a field study in order to contribute towards finding answers to a given research problem (Kothari 2004). Method is the techniques of data collection and analysis. Generally, research methodology consists of various types of research methods (Croom 2010; Karlsson 2016). This chapter presents the research design in Section 2.1, and the choice of research methods along with justification to answer the research methods in Section 2.2.

2.1 Research design

This section discusses the general approach followed in this study to answer the research questions and ultimately, achieve the research objective. This research is considered to be a deductive research. Deductive research, also referred to as top-down logic, begins with assumptions based on existing knowledge or literature, then research questions based on established theory or knowledge are formulated, then data is collected, and finally after the analysis, conclusions are drawn which leads to confirmation or rejection of the initially formed hypotheses (Karlsson 2016). The research design followed in this thesis is depicted below in the form of a flowchart in the figure 2.1. We begin the research with the problem description, then formulate research questions to achieve the research objective, then use two research methods for the qualitative and quantitative analysis. Literature review supported us in shortlisting the required forecasting models for the quantitative analysis. The data for the quantitative analysis was obtained from the case-company and the models were calibrated using the data. The models were also optimized using the grid search and hand-tuning techniques. They were finally compared against each other based on the performance metrics to find an appropriate forecasting method for short univariate time-series.

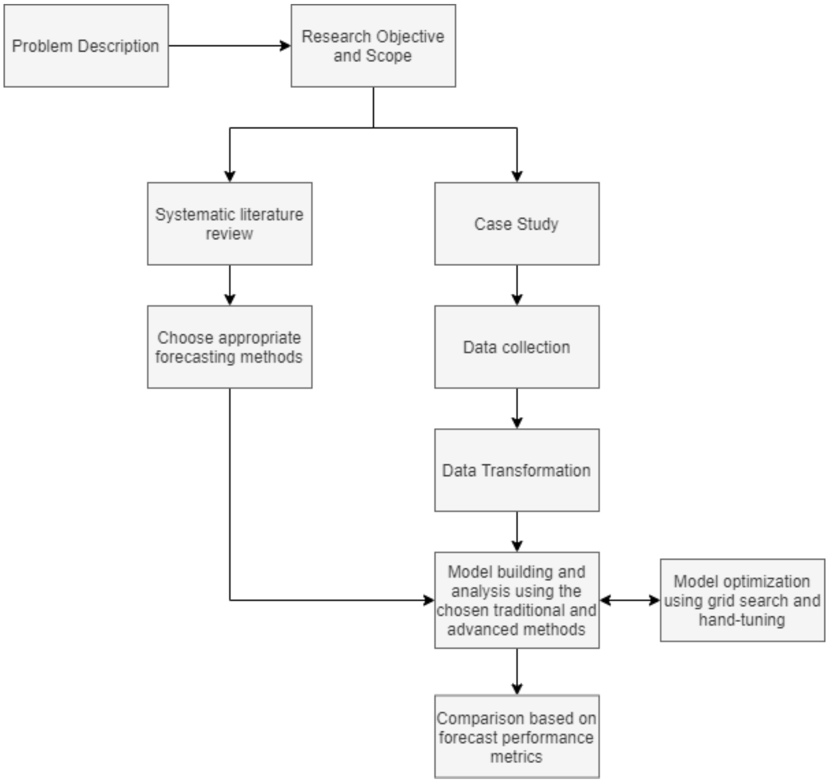

Figure 2.1: Research design

2.2 Research Methods

Research methods are of two types, quantitative or qualitative. A quantitative approach uses mathematical and statistical tools to manage the analysis of numerical data while a qualitative approach is concerned with analysis of data in textual form, and they are related with constructivism, interpretation and perception (Croom 2010; Karlsson 2016). This thesis is conducted as both qualitative and quantitative research, primarily systematic literature review was to conduct qualitative analysis and case-study was used to conduct quantitative analysis, because the focus is on both exploring and evaluating different forecasting methods.

The following sections explains the literature review and case-study approaches in detail.

2.2.1 Literature Review

Literature review is used as the primary method in this study and is conducted based on a systematic literature review approach. A systematic review establishes a firm foundation for future research and facilitates theory development, aligns existing research, and uncovers areas where additional research is needed (Buer et al. 2018; Webster and Watson 2002). A preliminary literature study was conducted to uncover state-of-the-art methods in theory relevant for the topic of research and to identify potential gaps in need of filling. The literature study aims to map existing knowledge and ideas on the chosen topic and discuss their strengths and weaknesses (J. W. Creswell and J. D. Creswell 2017). It is essential to know if the defined problem has already been solved, and if not, understand the current status of the problem (Croom 2010; Karlsson 2016). Gaining general knowledge from the current literature contributed to establish scope and define the contribution of this research.

The systematic literature review is performed using a series of transparent steps which ensures replicability. The literature review is performed using the PRISMA process (Buer et al. 2018) to narrow down the search results and is depicted in the figure 2.2. The goal of the study was to investigate the most suitable forecasting method for medium-term forecast horizon based on short time-series data and the relevant evaluation metrics required for measuring the performance of the forecasting methods. The process was conducted in the following manner:

1. The articles were identified using the keywords with Boolean operators (building block search) in different databases. The duplicate articles were removed by comparing the articles collected across different databases.

2. During the screening process, the titles and abstracts of the articles were investigated and excluded articles that were non-English articles, not related to the topic, not peer-reviewed and unavailability of the full text. This filtration leads to a collection of unique articles that require further examination. Only journal articles, book chapters and conference reviews from the year 2000 were considered to stick to recentness.

3. During the eligibility process, the articles were read thoroughly, and the vaguely related articles were excluded. This gives the final set of articles that must be included in the study.

4. Some of the relevant articles were also found through the cited-reference search (also called as ‘snowball effect’) method. The references section and introduction section of relevant articles were briefly scanned to find more literature papers. Some of the databases (mentioned later) included a cited-by function which supported us in finding articles related to the particular examined article more efficiently. This method was used to find some articles related to some particular topics and issues.

Potentially relevant articles were always checked for the number of citations because it helps in identifying the validity and degree of acceptance of the method presented in the article; however, for the articles published in the recent few years, the number of citations is not considered as a critical factor because the lead-time of the journal articles is around 8-14 months, and it takes time to appear as a reference (Karlsson 2016).

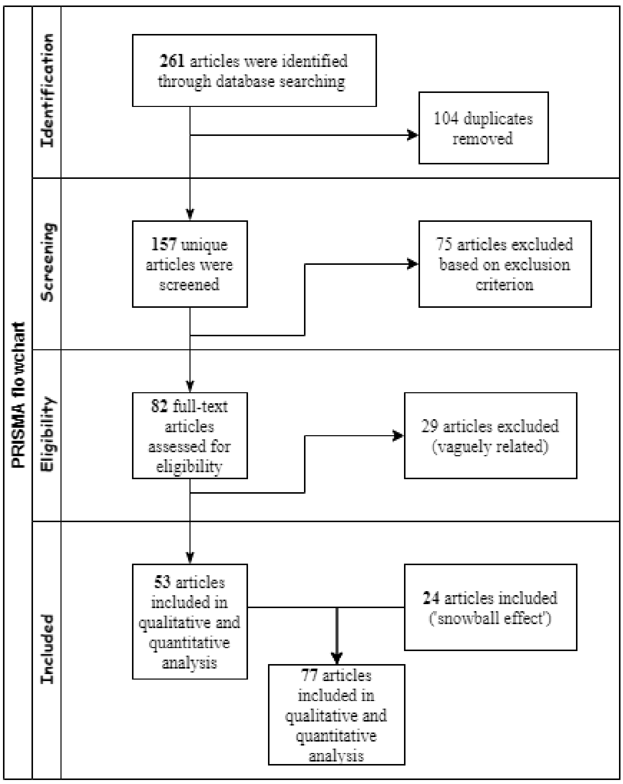

Figure 2.2: PRISMA process flowchart

Literature review was conducted to determine the various forecasting techniques available and to assess their applicability and performance on the given time-series data set. The keywords used for the literature search is present in table 2.1. Various combinations of the keywords and also their synonyms were used to find the relevant articles. These articles will help us in selecting the forecasting techniques for our analysis and consequently understand their performance on the provided data set. Relevant literature was mainly found by searching in the

Block 1	Block 2	Block 3
time series forecasting	univariate	statistical
sales forecasting		traditional
demand forecasting		machine learning
		artificial intelligence

Table 2.1: Keywords used for literature search

databases Scopus, Web of science, Google Scholar, ProQuest, Springer, blogs from Medium, and NTNU's library search engine Oria. The reference management software, Zotero, has been used to manage references throughout the project. The references have mainly been imported to Zotero by using the zotero plug-in in the web-browser. The plug-in helps in taking quick snapshots of the webpage or article or book and automatically inputs all the required details in the respective rows. Some of the citations were also directly copied from the databases in the Bibtex format. All the citations were exported to bibtex format and then were used in Overleaf. an online latex editor.

2.2.2 Case study

The secondary method which is used in the research is an exploratory case study. A case study is a history of a past or current phenomenon, drawn from multiple sources of evidence (Karlsson 2016; Voss et al. 2002). It is particularly useful in development of new theory. Meredith (1998) also claimed the same, 'since the explanation of quantitative findings and the construction of theory based on those findings ultimately have to be based on qualitative understanding, case research is very important for theoretical advancements in the operations management field'. The reasons for choosing exploratory case study as our research method are:

1. It helps in the development of new theory as stated above based on the patterns and relationships between the key variables. It is relevant to our study as we are trying to find an appropriate forecasting method for a short univariate time-series.

2. It is useful to obtain real-life data and achieve better comprehension of the nature and complexity of the operations and events taking place.
3. The applicability of the newly developed theory is valued by the practitioner because theory-building is interlined with empirical evidence. The theory is analysed and compared with the obtained empirical data.
4. It helps in answering the research questions such as 'what' and 'how'. In our study, it helps in partly answering RQ1.

Data collection

The data was collected using a combination of methods which include semi-structured interviews (which were guided by questionnaires and informal conversations) and retrieval of historical demand data from the database which will serve as an input to the forecasting models. Semi-structured interviews was chosen as a data collection method because of its versatility (Karlsson 2016). Semi-structured interviews were used to understand the business processes involved in demand forecasting and to validate the primary data exploration. Data summaries and visualizations were presented to business experts to check the correctness of the collected data.

The empirical data for the quantitative analysis was retrieved from the project delivery data of cage-P. Since the customer receives the product only on the scheduled delivery date, we have considered those dates as the actual demand date. Hence, the project delivery data is considered as the historical demand data in this study. The historical demand data was recorded on an weekly basis in an excel sheet, but we have aggregated them to quarterly basis as requested by the company. We have only considered their best selling circular fish farming cage, cage-P, for the analysis because it was requested by the company and it formed 80 percent of the total production of fish farming cages in a year. The time-series data was explored for missing values, duplicate values, and outliers. Furthermore, data visualizations were created to understand the trend, problem, and patterns in the time-series.

Data analysis

The quantitative analysis was conducted on the historical demand data of the case-company. The literature review supported in narrowing down the required forecasting methods and the chosen methods were used in the quantitative analysis. The results of the analysis were validated using the literature. The results was also presented to the case-company to further validate the obtained results. The software environment adopted for the data analysis was Python programming language and Jupyter notebook (was utilized as the editor). Python is a general-purpose programming language (unlike R and Matlab) that is easy to use due to its focus on readability. It is widely used in data analysis owing to the exceptional library support and offers a huge collection of data visualization libraries (Brownlee 2017). Libraries that are mainly used in our data analysis are Pandas (McKinney et al. 2010), Numpy (Harris et al. 2020), Statsmodels (Seabold and Perktold 2010), sci-kit learn (Pedregosa, Gaël Varoquaux, et al. 2011) and Keras (Chollet et al. 2018). Visualization library that was primarily used in the study was Matplotlib (Hunter 2007).

Chapter 3

Case study: Scale AQ

This chapter presents the case company involved in the study. Section 4.1 introduces the company and its operations. Section 4.2 discusses about its supply chain, and finally section 4.3 discusses the current demand forecasting procedure that is used.

3.1 General information

Scale AQ is the case company involved in this study. The company is a global supplier of technology and infrastructure for land- and sea-based aquaculture. It has been in the aquaculture industry for over 40 years and is an integration of five companies, Steinsvik, Aqualine, AquaOptima, Pan Logica and Moen Marin. The company has around 850 employees world-wide and their offices are strategically located in 12 countries headquartered in Norway. The annual revenue of the company in 2019 was around two billion Norwegian Kroner. Their products ranges from infrastructure that is required for aquaculture such as sea-based/land-based cages, feeding stations and so on, and the technology required for monitoring the fishes such as the cameras, lighting, software, sensors and so on. We will be focusing only on their best selling sea-based cage, cage-P, in our study.

3.2 Supply Chain

A general overview of the company's supply chain is shown in figure 3.1 in the form of a control model. The main components required for the production of the cages are brackets, nets, pipes, walkways, and ropes. The suppliers of the components are spread globally and they usually deliver the components to the main warehouse at Frøya. The components are then transported (by truck or ship) to the other warehouses (Bømlo and Tovik) or the project sites for cage assembly. Trains are used as a mode of transportation by the company for rush-orders in order to reduce the transportation lead-time. The project sites are usually temporary spaces close to the customer's location that are rented for the project term and the cages are assembled on those sites. The brackets supplier and pipes supplier also deliver the respective components to the project site. The comprehensive process of cage production could be abridged to a production process where PE-pipes are welded together, and brackets are threaded onto them. Other components such as the walkways are then secured to the brackets. The mooring system aids to forestall the free movement of the cage in the water. The finished cage is launched into the sea, and is stored in a temporary mooring system which serves as a temporary storage site. Later, they are towed by the customer to their location. The finished cage is stored in the temporary storage and they are towed by the customer to their location. The cage net is equipped to the finished cage present at the customer's location. The customer order decoupling point is at assembly and the company follows make-to-order (MTO) production strategy. Their customers could be divided into two distinct groups, local customers within Norway and global customers. We focus on only the local customers in this study because of the uncertainty in demand.

This study focuses only on the brackets due to the long lead-time involved. As we can see from the figure 3.2, the replenishment lead-time of the brackets required for the cage production is long (around 15-18 weeks). The lead-time for the replenishment order involves the setup and planning time for production of brackets. Production lead-time is around 16-20 days to produce brackets based on the volume required. The longest lead-time (75-80 days) is involved in loading of containers to the vessel and transporting the brackets to the nearest port. Receiving the containers from the nearest port to the warehouse takes another 1-2 weeks. Owing to the long replenishment lead-time of the brackets, it was difficult to predict and meet the unanticipated surge in cage demand during the peak-production season (January-May) especially for the cages with short delivery lead-time. This problem could be resolved by a number of methods, and we

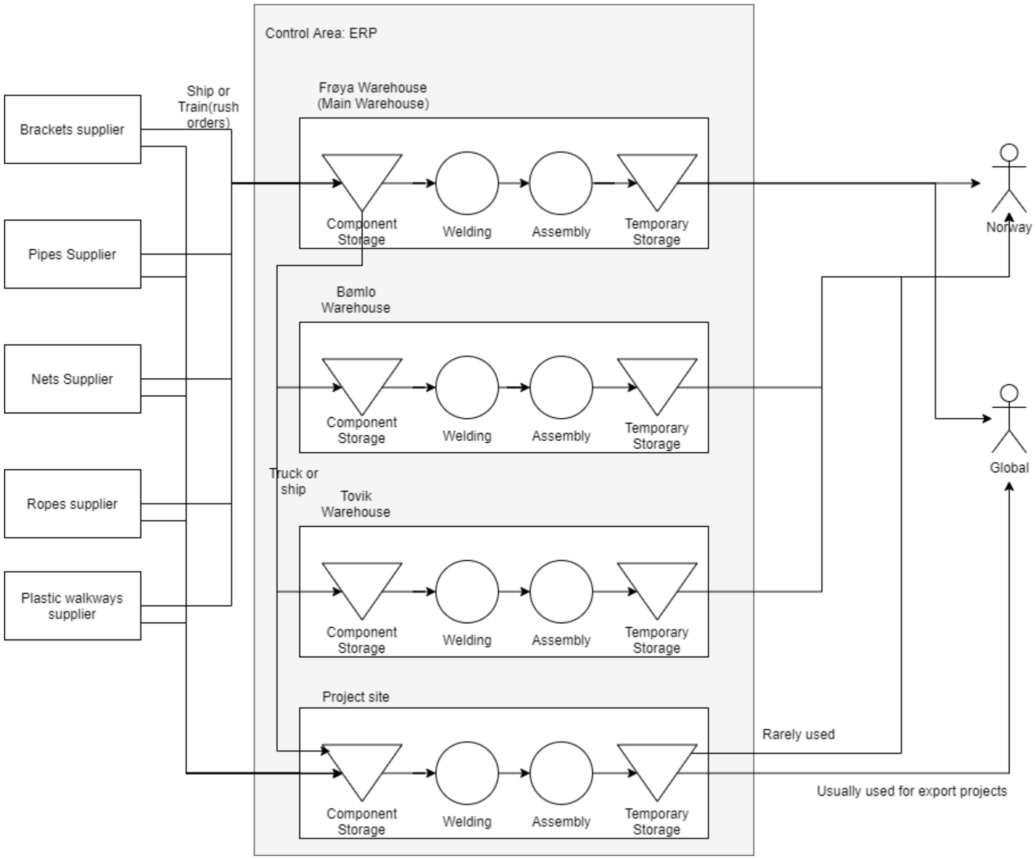

Figure 3.1: Control model of the supply chain of sea-based cage components

have chosen to improve the demand forecasts. The following section discusses about the demand forecasting practice used in the company.

3.3 Forecasting method

The company mainly relies on seasonal naive and judgemental forecasting for forecasting their demand. Seasonal naive serves as their statistical based forecast. Historical demand data is used for generating seasonal naive forecasts. Seasonal naive method is explained in detail in section 4.5. Judgemental forecasting are usually performed by using the Delphi method (William J. Stevenson 2014). The main stages involved in a Delphi method are as follows:

1. A panel of experts (usually head of each department such as Sales, Supply

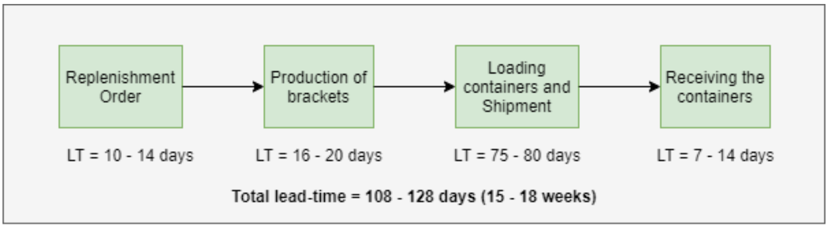

Figure 3.2: Replenishment lead-time for the brackets

chain, Production, Marketing, Finance and so on) are assembled

2. Forecasting tasks/challenges are set and distributed to the experts.
3. Experts return initial forecasts and justifications which are then compiled and summarised to provide feedback.
4. Feedback is provided by the facilitator (usually the chief operations officer) to the experts who now review their forecasts in the light of the feedback. The feedback usually consists of summary statistics of the forecasts and outlines of qualitative justifications. The process is iterated a number of times until all the experts reach a level of consensus.
5. Final forecasts are constructed by the facilitator by aggregating the experts' forecasts and using his domain knowledge and experience.

Chapter 4

Forecasting models

This chapter provides forecasting models that will be used for the quantitative analysis in the later chapter. The structure of the chapter is organised as follows: Section 4.1 provides a brief introduction to time-series forecasting; Section 4.2 presents the various forecasting methods; Section 4.3 compares the models qualitatively to find appropriate models for the quantitative analysis; Section 4.4 discusses the model selection criteria and presents the final shortlisted models for the quantitative analysis; and finally, Section 4.5 presents the detailed mathematical formulation of each model.

4.1 Time-series forecasting

A time-series is a sequence of observations recorded at successive equally spaced points in time (Deb et al. 2017; Rob J Hyndman and Athanasopoulos 2018). The data may be observations of demand, temperature, earnings, profits, shipments, accidents, productivity, and so on. Forecasting techniques developed for time-series data are based on the assumption that past values of the series is a good indicator of future values. This characteristic of time-series makes it a widespread problem attracting significant interest in research (Makridakis et al. 2018; Parmezan et al. 2019). Time-series analysis has been used for various applications such as econometric forecasting (Ahmed et al. 2010), quality and process control (Naim and Mahara 2018), sales forecasting (Brownlee 2017; Papacharalampous et al. 2018; Pavlyshenko 2019; Van Belle et al. 2021), health surveillance (Papastefanopoulos et al. 2020; Shih and Rajendran 2019; X. Zhang et al. 2014), energy demand forecasting (Divina et al. 2019; Jung-Pin et al. 2020; Martínez-Álvarez et al. 2015; Wang et al. 2018), etc. One of most renowned resource to study about time-series forecasting is a book by Rob J Hyndman and Athanasopoulos (2018). It is a free online textbook that helps aspiring practitioners to apply time-series forecasting in real-life problems.

While analysing a time-series data, the underlying behaviour of the time-series can be established by plotting the data and visually examining the plot. It is generally composed of the following five main components (Chopra and Meindl 2016; William J. Stevenson 2014):

1. **Level (L):** It refers to the scale of a time series.
2. **Trend (T):** It refers to a long-term increase, decrease or stagnation in the data.
3. **Seasonal (S):** It refers to short-term, quite regular fluctuations within a year according to the season of the year or time of the day. Human decisions (such as timing of price-promotions) can also cause seasonal behaviour.
4. **Cyclical (C):** It refers to wave-like variations of more than one year's duration, often related to factors such as economic or political conditions.
5. **Random variation (ε):** It refers to the residue that remain after all the other variations are accounted for.

Cyclical component is not useful for our analysis and hence it is not mentioned in the above models. Based on the major components of time-series, a time-series can be modeled into the following forms (Chopra and Meindl 2016; Rob J Hyndman and Athanasopoulos 2018):

- ***Additive model:*** It is most suitable if the magnitude of the seasonal fluctuations, or the variation around the trend, does not vary with the level of the time series. The components are independent of each other. It is of the form:

$$Data(t) = L + T + S + \varepsilon \tag{4.1}$$

- ***Multiplicative model:*** It is most suitable if the magnitude of the seasonal fluctuations, or the variation around the trend, appears to be proportional to the level of the time series. The components are not necessarily independent of each other, they could impact one another. It is of the form:

$$Data(t) = L \times T \times S \times \varepsilon \tag{4.2}$$

- ***Mixed model:*** Sometimes a combination of both additive and multiplicative model is used. One example of such a decomposition be:

$$Data(t) = ((L + T) \times S) + \varepsilon \tag{4.3}$$

4.2 Forecasting methods

"Prediction is very difficult, especially if it's about the future."

–Nils Bohr, Nobel laureate in Physics

This quote reveals the importance of validating an out-of-sample forecast. Usually, it is often easy to come across a model that fits the historical data well but identifying a model that correctly discovers the patterns in the historical data that will continue to remain in the future is troublesome.

The method of time-series forecasting can be broken down into two simple steps. The first step is understanding the data which involves obtaining the structure and identifying the underlying intrinsic patterns of the observed data. The second step is the model fitting which involves fitting a mathematical model to the time-series data in order to make future predictions. This step is usually a complex and challenging part in time-series forecasting (Parmezan et al. 2019). Generally, there are two types of time-series analysis:

1. **Univariate time-series analysis:** It is a time-series containing record of a single time-dependent variable, that is, only one variable will vary with time. For example, temperature of a place recorded on a hourly basis. To predict the future values, only the past values of the observation is used.

2. **Multivariate time-series analysis:** It is a time-series containing record of multiple time-dependent variables. The variables are dependent on each other along with time. For example, humidity, wind speed, and cloud coverage are also recorded on an hourly basis along with temperature. To predict the future values of temperature, the past values of temperature as well as the values of humidity, wind speed, and cloud coverage are considered.

In our study, we consider only univariate time-series analysis. Based on the literature of De Gooijer and Rob J. Hyndman (2006), Rob J Hyndman and Athanasopoulos (2018), Papacharalampous et al. (2018), Papastefanopoulos et al. (2020), Parmezan et al. (2019), and Wang et al. (2018) and Jung-Pin et al. (2020), we have identified the following methods: Naive, Moving Average, Exponential

Smoothing (ES), Autoregressive Integrated Moving Average (ARIMA), Facebook Prophet, Random Forests, Extreme Gradient Boosting (XGBoost), Support Vector Regressor (SVR) and Neural Networks (NN). These methods are categorised into two groups: Traditional forecasting methods and Advanced forecasting methods. Traditional forecasting methods consists of Naive, MA, ES, and ARIMA. While the advanced forecasting methods, which uses the principles of machine-learning and artificial-intelligence, consists of the remaining mentioned models, that is Facebook Prophet, Random Forests, XGBoost, SVR and NN. The hierarchy of time-series forecasting methods can be seen in figure 4.1. The following sub-sections briefly describes the aforementioned models in detail.

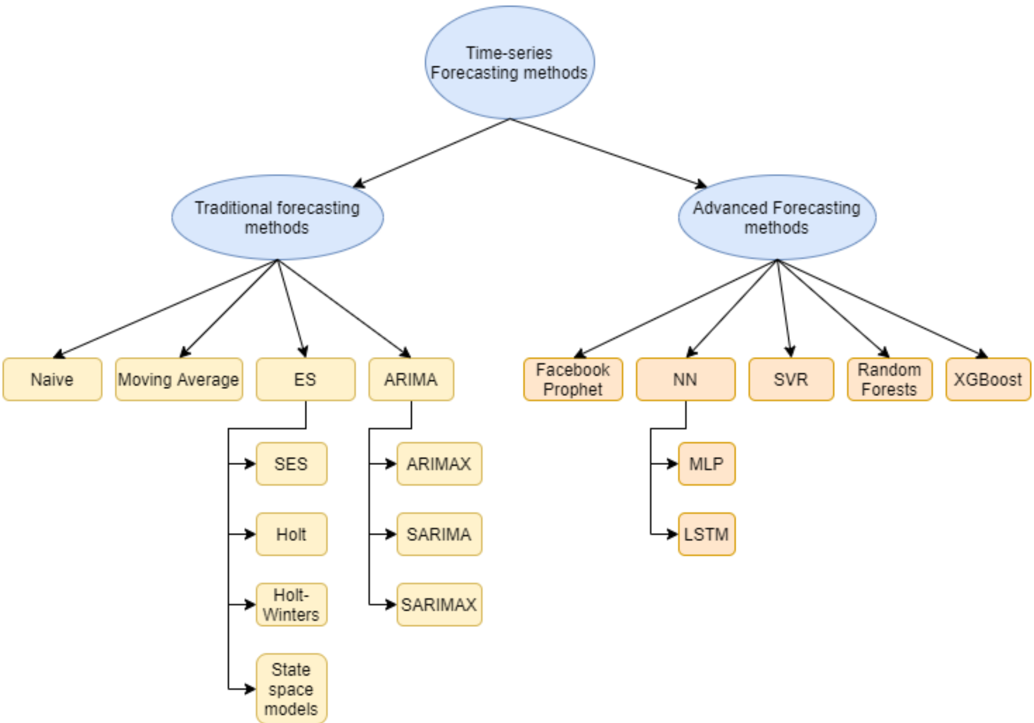

Figure 4.1: Hierarchy of time-series forecasting methods

4.2.1 Traditional Forecasting methods

Traditional forecasting methods are based on modeling and extrapolating the past time series structure into the future. Traditional time-series forecasting methods have been used extensively in industries since many years and the two models that have dominated in practice are Exponential smoothing (ES) and ARIMA (De

Gooijer and Rob J. Hyndman 2006; Rob J Hyndman and Athanasopoulos 2018). Makridakis et al. (2020) claimed that these two models were used as 'standards for comparison' in the recent M4 forecasting competition due to its widespread use in practice and relatively good forecasting accuracy.

Naive

It is most basic and simple method for forecasting. Each forecast value is equal to the value of the last observation in the time-series, that is, forecast value at time t is equal to the value observed at time $t-1$. This method is also known as **random walk forecasts** because it is best used when the time-series follows a random pattern. For time-series with seasonal behaviour, we use a variant of naive method called **seasonal naive**. In seasonal naive, each forecast value is equal to the last observed value from the same season of the year (e.g., the same month/quarter of the previous year) (Rob J Hyndman and Athanasopoulos 2018). If m is the seasonal period (for quarterly observations $m = 4$, for monthly observations $m = 12$, etc), the forecast for value Z at time t will be the last observed value at time $t - m$. Formally, it could be written as:

$$Z_t = Z_{t-m} \tag{4.4}$$

For example, the forecast of future June observations is equal to last observed June observation. This model performs very well when the time-series consists of only random fluctuations with seasonality. This model is also used as a base model to benchmark the performances of new forecasting models (Brownlee 2017; De Gooijer and Rob J. Hyndman 2006).

Moving Average

This is another simple technique that is used in practice widely. The forecast values are predicted by taking an arithmetic average of the last r values of the time-series. It is denoted as MA(r), where r stands for the number of observations included in the average. For example, MA(3) is the average of the last three observation values. Formally it is written as:

$$Z_{t+1} = \frac{Z_t + Z_{t-1} + \cdots + Z_{t-r+1}}{r} \tag{4.5}$$

In the above equation, Z_{t+1} is the forecast value at time $t + 1$ which is obtained by performing an arithmetic average on the last r observation values. The higher the value of r, the more uniform (smoothed) will be the predicted data behaviour.

The main drawbacks of this model is their disability to handle with the trend and seasonal components of a time-series and insignificance of the recent observations due to equal weights assigned to all the observations (De Gooijer and Rob J. Hyndman 2006; Parmezan et al. 2019).

Exponential Smoothing (ES)

The exponential smoothing models generate forecasts by taking weighted average of the past observations with the weights decaying exponentially over time (Akpinar and Yumusak 2016; Papastefanopoulos et al. 2020). The trend and seasonality components are captured in this model using smoothing parameters (De Gooijer and Rob J. Hyndman 2006). There are mainly three types of exponential smoothing models, simple exponential smoothing (SES), Double exponential smoothing or Holt's linear trend exponential smoothing (Holt), and Holt-Winters exponential smoothing (HW).

SES is similar to MA where the most recent observation gets the highest weight and it decreases exponentially over time. Thus, the most recent observation has a higher influence on the predicted value than the previous observations. The Holt model is an extension of SES where an additional parameter is added to capture the trend. Similarly, smoothing parameters are added for the trend and seasonality in the HW method. The trend or seasonal components could be additive or multiplicative and the equations for modeling are chosen accordingly. Additive models are chosen when the trend or seasonal component change constantly over time while the multiplicative model is chosen when the trend or seasonal component change proportional to the level of the time-series. The reader could refer to Rob J Hyndman and Athanasopoulos (2018) for the detailed calculations.

Combining the model with the state space models (Rob J Hyndman, Anne B Koehler, et al. 2002) has improved their dependability in the statistical domain. In a state space model, the forecast is generated by factoring in the forecast error along with the three components of the time-series (that is, level, trend and seasonal). The state space models also generate prediction intervals along with point forecasts similar to the three exponential smoothing models aforementioned (Rob J Hyndman and Athanasopoulos 2018; Rob J Hyndman, Anne B Koehler, et al. 2002). The error term could be additive or multiplicative. Additive error is similar to additive trend and seasonality where the error change constantly over time while the multiplicative error change proportional to the level of the time-series. To differentiate between additive and multiplicative errors, the state-space models are usually labelled as ETS(.,.,.), where ETS here stands for Error, Trend and Seasonality. For instance, an additive model is denoted as ETS(AAA) where A's in the bracket stand for additive error, additive trend, and additive seasonality. The detailed calculations for each combination of the model could be referred in Rob J Hyndman and Athanasopoulos (2018).

Though ARIMA models have outperformed ES models in most of the instances, ES models have an advantage of not requiring data transformation (such as logarithm, Box-Cox or differencing) on some particular time-series. The performance of the ARIMA models is impacted by these data transformations. A few drawbacks of implementing this model are that the three ES models (SES, Holt and HW) do not consider the error terms, cannot reflect non-linear relationship, is sensitive to outliers and unusual events, absence of exogenous variables, and the performance of the model depends on choice of initial value and values of the smoothing parameters (Akpinar and Yumusak 2016; Papastefanopoulos et al. 2020). In the articles of Akpinar and Yumusak (2016), Ilbeigi et al. (2017), Kalekar (2004), Ramos and Oliveira (2016), Shih and Rajendran (2019), and J. Zhang et al. (2016), ES models performed best compared to the other models.

Auto-regressive Integrated Moving Average (ARIMA)

While ES models focus on the description of the trend and seasonal factors of the time series, ARIMA models focus on the correlation between the lagged time-series observations (Nau 2020). ARIMA models tries to capture the strong correlation of the present and past values that is generally present in the time-series data. The ARIMA models of order(p,d,q), that is ARIMA(p,d,q), is a combination of three operations: (i)autoregression(AR(p)) (ii) integration (d) (iii) moving average (MA(q)). Autoregressive (AR(p)) part of the ARIMA model captures the autocorrelation between the present and the past values. The Moving Average (MA(q)) part of the ARIMA model captures the past forecast errors of the model. The integration (d) part of the ARIMA model stands for degree of differencing. Differencing operation comprises of taking difference between consecutive observations. An important prerequisite for ARIMA models is that the time-series should be stationary (De Gooijer and Rob J. Hyndman 2006; Rob J Hyndman and Athanasopoulos 2018; Nau 2020). A stationary time series is a time series whose properties do not vary with time at which it is observed. Thus, a series with trend or seasonality is considered a non-stationary series. We can convert a non-stationary series to stationary series by differencing. Differencing is performed by taking a difference of the consecutive observations. Multiple statistical tests are also available to test the stationarity of the time series. Augmented Dickey-Fuller (ADF) test and the Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test are the most widely used statistical tests.

The ARIMA models are used to capture only the trend behaviour in the time-series. To capture the seasonal effect, we use an extension of ARIMA model called seasonal ARIMA, also known as SARIMA. Four additional parameters $(P, D, Q)_m$ are added to the model. (p, d, q) describes the non-seasonal part of the time-series and $(P, D, Q)_m$ describes the seasonal part. The parameters P and Q, similar to p and q, captures the seasonal autoregressive behaviour and seasonal MA behaviour. D is the seasonal differencing. m indicates the seasonal period.

The design of ARIMA and SARIMA model is described as a stochastic model building process. This model is built using iterative cycle of Box-Jenkins (Box et al. 2011):

1. *Selection of model:* The selection of model is based on the time-series characteristics. If the series shows only trend behaviour, then ARIMA model is preferred. Alternatively, if the series shows both trend and seasonal behaviour, then SARIMA model is preferred.

2. *Identification of model orders:* The parameters of the model (p, d, q, P, D, Q, and m) are set with the aid of correlograms or information criteria. They are chosen in such a way that it best describes the data. Correlograms comprises of autocorrelation and partial autocorrelation graphs which are visually inspected to select the model orders. Information criteria is also used to find the parameter orders and the Akaike Information Criterion (AIC) is most commonly used. AIC helps in determining the best model for a given time-series out of the given multiple models. The model with the lowest AIC is usually preferred. The more parameters a model has, better is the fit but the model might not have the lowest AIC. The reader could refer to Rob J Hyndman and Athanasopoulos (2018) and Nau (2020) for detailed model order estimation.

3. *Estimation:* The parameters of the model are trained using the time-series and the model coefficients are estimated.

4. *Diagnosis of the fitted model:* The obtained model is validated on the time-series data to check if it has represented all the data characteristics. In practice, the estimates of errors (residues) are analysed for autocorrelation. If there is no autocorrelation between the residues, then the model can be ex-

trapolated to the future, else the practitioner has to select a different model and repeat the identification,estimation and diagnosis steps.

Since the number of parameters are high, there is a possibility of large number of combinations and there are no general rules to select all of the parameters. The experience of the analyst and his perception plays a huge role in modeling process. Hence to avoid the tedious manual process of model selection, there is pmdarima (also known as pyramid-arima) library (G Smith 2020) built on Python which automatically finds the model parameters.

Both the models can be extended to multivariate analysis, ARIMAX and SARIMAX, which includes exogenous variables that enables the analyst to add external information. The drawbacks of this method are that it requires a stationary time-series as an input and cannot describe non-linear relationships well that is present in the complex real-world problems (Papastefanopoulos et al. 2020; Wang et al. 2018). In the articles of Claveria and Torra (2014), Jere et al. (2017), Naim and Mahara (2018), Naim, Mahara, and Khan (2020), Padhan (2012), Shih and Rajendran (2019), J. W. Taylor (2008), Tularam and Saeed (2016), and Udom and Phumchusri (2014), ARIMA and SARIMA models provided superior prediction results over the other models.

4.2.2 Advanced Forecasting methods

Advanced forecasting methods were built mainly using machine learning algorithms. Machine learning methods unlike the traditional methods could describe the data properties without prior knowledge of their distribution (Parmezan et al. 2019). These models are flexible and show reliable performance when applied to complex and non-linear series because they are not dependent on the parameters to model the phenomenon's behaviour. Machine learning is the concept where a computer program has the ability to automatically learn from the provided training data by building a mathematical model and adapt to a new data instead of being explicitly programmed. Commonly, machine learning methods are classified as one of the following: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. We have chosen supervised learning method in this study. Supervised learning is a method where we use labeled data to optimize the model. In simpler words, we determine a predictive model using data points with known outcomes.The commonly used state-of-the-art advanced forecasting methods are discussed briefly in the following sub-sections.

Neural Networks (NN)

Neural networks (NN) are versatile methods for forecasting applications due to their capability of modeling non-linear complex problems as well as linear problems (Remus and O'Connor 2001; Guoqiang Peter Zhang 2001). NN is a machine learning method that is inspired by the information processing performed by the human brain (Haykin et al. 2009). The main logic behind an NN mathematical model is that the inputs get filtered through one or more hidden layers with hidden neurons before they reach the output neuron. Each neuron sums the weighted inputs and transfers the input through an activation function in order to produce a result. They are categorised into feed-forward neural networks and feedback neural networks. In feed-forward neural networks the neuron-to-neuron signals flow in only one direction, layer-by-layer. They are generally popular in time-series forecasting.

Neural networks can learn and generalize from the provided historical data about the patterns in the data-set. The training of the network is done with the idea of reducing the squared difference between the measured output and those predicted by the ANN model (Deb et al. 2017). The proposal of backpropagation learning algorithm, which is based on the reduction of the result error by setting the correct 'weights' combination, allowed NN with more than two layers. One such NN with multiple layers and that uses backpropagation algorithm is the Multi-Layer Perceptron (MLP). The MLP is the most widely used NN (Carbonneau et al. 2008; Deb et al. 2017; Parmezan et al. 2019). An MLP can have one or more layers of neurons between the input and output layers to conduct the input-output mappings. The inputs and the activation function in the neuron are assigned with weights which decides how the sum of weighted inputs will be mapped to outputs. The hidden layers aid to increase the computational power of the MLP network (Carbonneau et al. 2008). The structure of MLP with one hidden layer is shown in figure 4.2. The number of hidden layers, the number of neurons, and the activation function are generally dependent on the input data and they are manipulated to optimize the results. The bias shown in the structure aims to correct the net value by increasing or decreasing it. Although they are superior in dealing with huge volumes of data and proper generalization, they usually are difficult to interpret due to their "black-box" nature. In the articles of Abdel-Aal (2008), Buxton et al. (2019), Delic (2019), Gonzalez-Romera et al. (2006), Ismail Fawaz et al. (2019), Sharifzadeh et al. (2019), and Weytjens et al. (2019), MLP outperformed other models in predicting demand.

A recurrent neural network (RNN) is a feedback neural network where there are

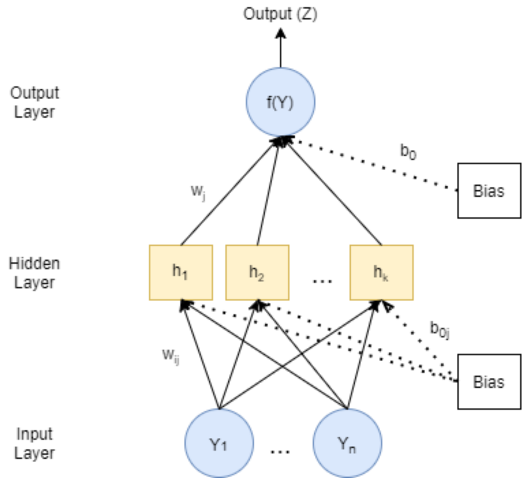

Figure 4.2: Structure of MLP with one hidden layer (adapted from (Parmezan et al. 2019))

cycles or feedback connections among neurons. Outputs from some of the layers of a recurrent network can be directly fed back as inputs to the same layer or previous layers generating dynamic feedbacks on errors of past patterns (Carbonneau et al. 2008). In this sense, recurrent networks can model richer dynamics than feedforward networks just like linear autoregressive and moving average (ARMA) models that have certain advantages over autoregressive (AR) models. Long Short Term Memory (LSTM) is a form of recurrent neural network and lately it has been gaining attention in the field of time-series forecasting (Brownlee 2018; Jung-Pin et al. 2020). The control flow in an LSTM is similar to an RNN, but the difference lies in the operations within the LSTM cells. The structure of LSTM and a LSTM cell is shown in figure 4.3. The LSTM cells contain various gates; input gate, output gate, and forget gate. The cell state (c_t) holds the memory of the network and transfers relative information to the next cell in the sequential chain. As we go down the chain, information is added or removed to the cell state via the gates which decides the information that is allowed to the cell-state. h_t is the output from the previous hidden layer,Y_{t+1} is the input data and Z_{t+1} is the output. σ and $\tanh$ are the sigmoid and $tanh$ activation functions. $\times$ and $+$ are the point-wise multiplication and addition operations. The gates learn from the input data and decides on the relevant information to be retained during the training (Bouktif et al. 2018; Phi 2020). This prevents the network from vanishing gradient problem (the gradient of the initial layer is exponentially decreased due to the large number of intermediate layers and thus the initial layer has negligible impact on the predicted result) and retain the long-term temporal information (Phi 2020). The activation functions inside the cell and number of LSTM cells can be varied. There is no global best configuration because the parameters of NN depends on the provided time-series.The drawbacks of NN are the interpretability of the model, overfitting and wrong predictions due to insufficient data, and the complex nature of parameterization of the models. In the articles of Abbasimehr et al. (2020), Bandara et al. (2019), Bouktif et al. (2018), and Weytjens et al. (2019), LSTM outperformed other models in predicting demand.

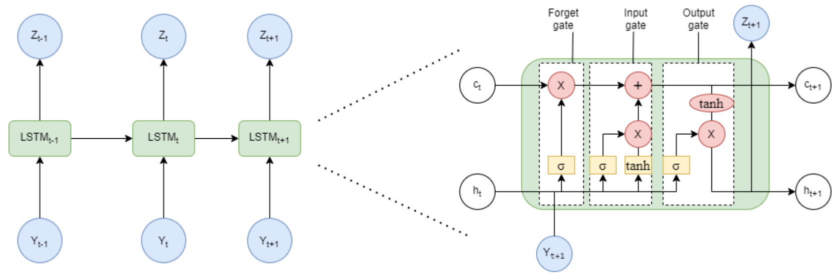

Figure 4.3: Structure of LSTM and LSTM cell (adapted from (Olah 2015))

Facebook Prophet

Generating high-quality and reliable prediction results with traditional methods required analysts with high statistical knowledge and experience. Due to the rare availability of such analysts, Facebook developed an open-source forecasting tool, known as Prophet, that would help an amateur with domain knowledge to generate reasonable forecasts with few easy-to-tune intuitive parameters. Prophet was initially developed to handle issues at Facebook such as predicting user activities. This method is designed to handle the common features of business time-series such as trend, seasonality, and so on, and is available in both R and Python (S. Taylor and Letham 2018).

It uses a decomposable time-series model comprising of three main components: trend, seasonality, and holidays. In its core, it is an additive regression model which has interpretable parameters that usually fits well with its default values. It uses a curve-fitting technique for the time-series fit and is fitted automatically using the Stan code (Carpenter et al. 2017) which considers the three main components mentioned previously. It allows the user to pick all the components that are related to the forecasting problem intuitively and effortlessly make the required changes by tuning the parameters. It works very well on a time-series showing multiple seasonal effects in the historical data. It is robust to missing data and shifts in trend, and typically handles outliers well (S. Taylor and Letham 2018).

It uses a framework called ”Analyst-in-the-loop” approach which is shown in the figure 4.4. The analyst-in-the-loop approach aims to combine the advantages of

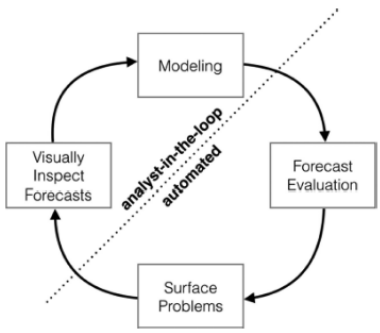

Figure 4.4: Analyst-in-the-loop approach (S. Taylor and Letham 2018)

both statistical model and judgemental forecasts by allowing the analyst with limited statistical knowledge to conduct automated model-fitting and use his domain knowledge and effort to improve the forecasts, when necessary, by including additional information.

Prophet utilizes two models to forecast trend, a saturating growth model and a piece-wise linear model (S. Taylor and Letham 2018). A model similar to population growth models in natural ecosystems is adopted, where there is non-linear growth that reaches a saturation point at a carrying capacity, for growth prediction problems. For non-saturating growth forecasting problems, a piece-wise model of constant growth-rate serves an useful alternative. The changes in the trend are automatically detected by Prophet using changepoints from the time-series data (S. Taylor and Letham 2018). The seasonality effect is modelled using Fourier series. The holiday effects are assumed to be independent and are considered in the model-building activity based on a predefined user-input list that consists of past and future holiday events.

The drawbacks of the model are that it is difficult to include other features than only seasonality or special events, model interpretability, manual tuning of the model is required when there are unexpected changes in the data structure, and longer computational time while predicting large number of variables simultaneously. There were only few articles that used Prophet in time-series forecasting and among those articles, in the articles of Papacharalampous et al. (2018), Samal et al. (2019), and Yenidoğan et al. (2018), Prophet performed better than the other models.

Support Vector Regression (SVR)

Support Vector Machines (SVM) are renowned classification machine-learning algorithm which categorises the data accurately. SVR is an extension of Support Vector Machines (SVM) which works by transforming the input data into a high-dimensional feature space by linear or nonlinear mapping and then forecast unknown data based on trained model (Xu et al. 2019). The model tries to minimize the maximum margins on the either side of the hyperplane thereby finding a narrowest tube that comprises the most amount of training data (Awad and Khanna 2015). The figure 4.5 shows the technique where ϵ is the size of the margins, $f(x)$ is the hyperplane and ξ represents the slack variables which stores the distances of the data not included inside the margin. SVR performs the non-linear regression using the so-called *kernal trick* by use of a kernal function, weight factor and regularization parameter (Awad and Khanna 2015; Santamaría-Bonfil et al. 2016). Linear, polynomial and Radial Basis Function (RBF) kernal functions are the most widely applied in practice (Parmezan et al. 2019). In the articles of Kusiak et al. (2009), Parmezan et al. (2019), Samsudin et al. (2010), Santamaría-Bonfil et al. (2016), and X. Zhang et al. (2014), SVR outperformed the other models that were considered in the study. The drawbacks of this model

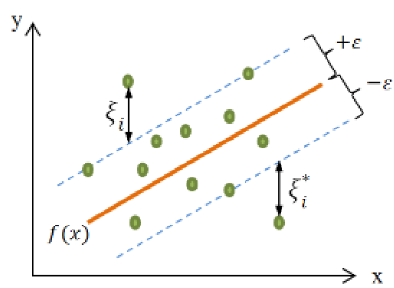

Figure 4.5: An example of linear support vector regression (Chanklan et al. 2018)

are that it is sensitive to missing data and cannot be applied on very large data-sets (Wang et al. 2018).

Random Forests

Random Forest regression is an ensemble learning method that uses decision trees for regression. Ensemble learning method is a technique that combines predictions from multiple models to make a more accurate prediction than a single model. It was first proposed by Breiman (2001). The figure 4.6 shows the structure of a Random Forest. We can notice that the trees run in parallel with no interaction amongst them. Each tree is built from a randomly selected training subset which also includes the features subset. A Random Forest operates by constructing several binary decision trees fitted using bootstrap samples during training time and the output is output is obtained by aggregating over the ensemble, that is the mean of the prediction of all the trees (Dudek 2015). This method is called bagging which reduces the variance and overfitting issue thereby improving the accuracy.

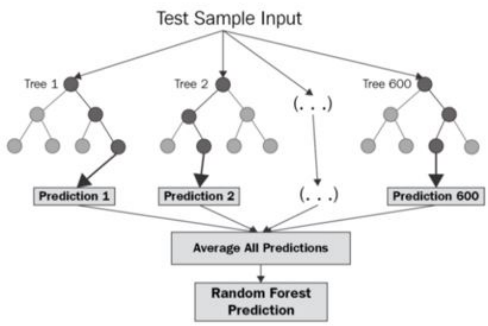

Figure 4.6: Structure of Random Forest (Bakshi 2020)

The drawbacks are model interpretability, requires longer training time and uses a lot of memory for very large data-sets (Divina et al. 2019). In the articles of Johannesen et al. (2019), Kim et al. (2019), and Lahouar and Ben Hadj Slama (2017), the RF model performed the best among the other models.

Extreme Gradient Boosting

Contrary to tree bagging methods such as RF, the prediction of different trees are sequentially pooled in tree boosting methods. Gradient boosting is a method whereby trees are built one at a time, where each new tree in the sequence tries to correct errors made by the previously built tree (that is it combines a number of weak learners to form a strong learner). Extreme gradient boosting, also popularly known as XGBoost, is a decision tree-based ensemble learning algorithm that uses a gradient boosting framework. The strength of this method is mostly due to its scalability and high predictive power. Some of the features of XGBoost (Chen and Guestrin 2016) are as follows:

- XGBoost is almost ten times faster than the other boosting techniques because of parallel learning and distributed computing.
- It also includes a variety of regularization (L1 and L2) which reduces overfitting and it improves the overall performance. Therefore, it is also called as *'regularized boosting'* technique.
- It can manage sparse data by using sparsity-aware split finding algorithm.
- It can handle weighted data efficiently by using distributed weighted quantile sketch algorithm.
- It uses the hardware resources optimally by system cache optimization.
- It employs out-of-core computation which enables a basic computer to process massive datasets that do not fit in the memory.

XGBoost model has been chosen as a winning solution in many machine learning competitions such as Kaggle (Chen and Guestrin 2016). In the articles of Demolli et al. (2019), Divina et al. (2019), Tanizaki et al. (2019), and Torres-Barrán et al. (2019), XGBoost model was proved to be superior to the other models in predicting wind energy demand and electric energy consumption respectively. An excellent resource for XGBoost is its official documentation page (Chen, He, et al. 2018). The drawbacks of this method are that it requires large data-set to perform well, model interpretability and finding optimal hyperparameters for the model.

4.3 Qualitative comparison of models

In this section, each of the model is qualitatively compared so that it aids in the selection of models for quantitative analysis. Though most of the literature addressed energy demand forecasting issues, we assumed that we could adapt the models to our problem based on their results. We have summarised the models in table 4.1 and table 4.2 with its features, advantages, disadvantages, number of times it has performed the best in the literature, and finally its applicability to the type of forecasting horizons. Most of the articles reviewed in the study contained at least one traditional model and advanced model to understand which

Models	Feature	Advantage	Disadvantage	Best performed in literature	Forecast Horizon
Naive	Simple approach where the value of dependent variable is equal to the last observed variable. Usually used as a base model while benchmarking.	1. Useful for randomly fluctuating time-series problems such as stock-price prediction. 2. Can work with small data-points.	1. Produces poor results compared to the other sophisticated models 2. Not suitable for non-linear problems	-	Short-medium
Moving Average	Dependent variable is equal to the average of the past values	1. Able to cope up with trend 2. Easy to calculate	1. Produces poor results compared to other models 2. Not suitable for long-term forecasts and non-linear problems 3. Cannot explain seasonality	-	Short-medium
Exponential Smoothing	The most recent observations influence the forecasts by using exponentially decreasing weights.	Can explain trend, seasonality and random behaviour of the time-series	1. Not suitable for long-term 2. Cannot reflect non-linear relationships 3. Cannot include exogenous variables	6	Short-medium
ARIMA	Regression of the lagged values of the dependent variable and lagged values of the forecast errors.	1. Can explain a wide variety of time series and its components using its lagged values and forecast errors. 2. Can include exogenous variables	1. Requires stationary time-series 2. Cannot reflect non-linear relationships and not suitable for long-term. 3. Finding optimal values of parameters is challenging.	9	Medium-long

Table 4.1: A summary of several models commonly used in traditional methods

model is best suited for our study. We can see from the tables 4.1 & 4.2, only ARIMA models are capable of producing medium-long term forecasts compared to advanced methods where all the models were able to generate medium and long term forecasts. Also, we can notice that ARIMA and Neural networks performed the best the most number of times in the literature while Prophet being the least.

We found that traditional methods outperformed advanced methods in the literature of Claveria and Torra (2014), Makridakis et al. (2018, 2020), Naim and Mahara (2018), Papastefanopoulos et al. (2020), Peng and Chu (2009), Shih and Rajendran (2019), and J. Zhang et al. (2016), while the contrary in Abbasimehr et al. (2020), Abdel-Aal (2008), Ahmed et al. (2010), Carbonneau et al. (2008), Divina et al. (2019), Johannesen et al. (2019), Parmezan et al. (2019), Sharifzadeh et al. (2019), Weytjens et al. (2019), and X. Zhang et al. (2014). Papacharalampous et al. (2018) addressed that both advanced and traditional methods performed well in his article. However, most of the articles did not consider optimizing the traditional methods because the focus was on developing the advanced methods. Though machine learning models are known for overfitting when short time-series is involved (Buxton et al. 2019; Ismail Fawaz et al. 2019), we see that that they have performed better than the traditional methods in the articles of concurrent one Abbasimehr et al. (2020), Abdel-Aal (2008), Buxton et al. (2019), Cankurt and Subaşi (2016), Delic (2019), Rivero et al. (2017), Tanizaki et al. (2019), Xu et al. (2019), and Yu et al. (2018). Since the literature review does not indicate an appropriate method, we have therefore decided to test both the

Models	Feature	Advantage	Disadvantage	Best performed in literature	Forecast Horizon
Prophet	Easy to use automated forecasting tool without requiring statistical expertise	1. Can handle missing data and outliers well. 2. Intuitive parameters that can easily be tuned. 3. Easy to capture information regarding holidays or special events.	1. Difficult to include additional features other than seasonality and special events. 2. Model interpretability due to automated process. 3. Slow computing speed when predicting large number of variables simultaneously	2	Medium
Neural Networks	Imitates the information processing of the human brain neural network	1. Provides self-learning function and high-speed search for optimal solution. 2. Can learn and adapt to unknown systems 3. Can approximate any arbitrary complex non-linear relationship	1. Model interpretability 2. Hyperparameter tuning is a burden 3. Insufficient data can lead to wrong predictions	9	Medium-long
Support Vector Regression	Finds the best compromise between the complexity of the model and the learning ability based on the limited sample information	1. Can handle non-linear relationships using kernal function 2. Good for long-term time series 3. Good fitting and generalization	1. Lack of transparency of the results 2. Hyperparameter tuning is a burden 3. Sensitive to missing data and difficult to implement on very large datasets.	4	Medium-long
Random Forests	Prediction is based on the aggregate of the predictions of all the decision trees	1. Can handle non-linear relationships 2. Can prevent overfitting. 3. Can handle missing values.	1. Long training time for large data-set 2. Uses a lot of memory for very large dataset 3. Model interpretability	3	Medium-long
Extreme Gradient Boosting	Sequential correction of previous forecast errors. Highly scalable and good accuracy.	1. Hardware optimization 2. Regularization 3. Fast computing speed 4. Can handle sparse data and weighted data.	1. Model interpretability 2. Data hungry 3. Hyperparameter tuning is a burden	4	Medium-long

Table 4.2: A summary of several models used commonly in advanced methods

methods in our study.

Table 4.3 presents a summary of the articles that utilizes short time-series for forecasting. It comprises of the best forecasting model that was found in the particular article, the frequency at which the observations were recorded, the number of observations (or data-points), the presence of trend and seasonality in the raw data, the forecast horizon that was used in the literature, and finally, the forecast accuracy metric that was used for evaluation in the literature (this is explained in detail in section 3.4). We noticed that advanced forecasting methods performed well in most of the articles closely followed by traditional methods. Also, three of the articles (Delic 2019; Makridakis et al. 2020; Xu et al. 2019) had a combined model as the best performing model. We do not consider combined models in this thesis as it is beyond the scope of study.

4.4 Selection of models

From the literature study, we have shortlisted a number of methods. To enable appropriate comparison, we have chosen articles that included atleast one tra-

Author	Best Model	Frequency	Datapoints	Trend	Seasonality	Forecast Horizon	Forecast Accuracy Metric
Abbasimehr et al. (2020)	LSTM	Month	132	Yes	Yes	Medium	RMSE and sMAPE
Abdel-Aal (2008)	MLP	Month	94	Yes	Yes	Medium	MAPE
Buxton et al. (2019)	MLP	Quarter	77	Yes	Yes	Medium	MAPE
Cankurt and Subaşi (2016)	SVR	Month	180	No	Yes	Medium	RelMAE and RelRMSE
Claveria and Torra (2014)	ARIMA	Month	108	Yes	Yes	Medium	RMSE
Delic (2019)	Feed-forward NN	Day	70, 100	Yes	No	Medium	MAE, MSE, RMSE and MAPE
Makridakis et al. (2018)	ES and ARIMA	Month, year	126, 14	Yes	Yes	Medium	sMAPE and MASE
Makridakis et al. (2020)	ES-RNN	Month, quarter, year	42, 13, 16	Yes	Yes	Medium	sMAPE and MASE
Naim and Mahara (2018)	ARIMA	Month	34	Yes	No	Short	MSE, RMSE and MAPE
Papastefanopoulos et al. (2020)	ARIMA	Day	104	Yes	No	Short	RMSE
Peng and Chu (2009)	Classical decomposition	Month	48	Yes	Yes	Medium	MAE, RMSE and MAPE
Shih and Rajendran (2019)	ES and ARIMA	Week	35	Yes	Yes	Medium	MAE, RMSE and MAPE
Xu et al. (2019)	SARIMA-SVR	Month	157	Yes	Yes	Medium	MAE, RMSE and MAPE
Yu et al. (2018)	LSTM	Week	45	Yes	Yes	Medium	MSE

Table 4.3: A short summary of several models using short univariate time-series
[RMSE - Root Mean Squared Error, sMAPE - Symmetric Mean Absolute Percentage Error, MAPE - Mean Absolute Percentage Error, MASE - Mean Absolute Scaled Error, MAE - Mean Absolute Error, MSE - Mean Squared Error, RelMAE - Relative Mean Absolute Error, RelRMSE - Relative Root Mean Squared Error]

ditional method and one advanced method. We have chosen the most suitable models based on the criteria that is discussed in the followed sub-section.

4.4.1 Criteria

The criteria for the selection of models are:

1. The model should be the best performing model in atleast one literature article.
2. The model should be applicable to the short univariate time-series data adopted in the study.
3. The model should be capable of generating medium term forecasts
4. The model should be capable of handling trend and seasonality components in the time-series

4.4.2 Selection

Based on the literature review of forecasting methods discussed in the section 3.2, we came across the following models: Naive, Moving Average, ES, ARIMA, NN, SVM, Prophet, Random forests and XGBoost. Naive models are chosen because they are used by the company as the statistical model for forecasting. Therefore, they also act as a base model for the benchmarking. The seasonal variant of naive model, seasonal naive, is used in our analysis. Moving average is not considered in our analysis because it cannot handle the seasonality and also it has not performed well in atleast one literature. The Holt-Winters model and the state-space model in the exponential smoothing models are chosen because they can handle seasonality and also they have been the dominant models in six literature articles (can be seen in table 4.1). They can also generate medium term forecasts and can be used on short time-series. A seasonal variant of ARIMA, known as SARIMA, is considered in the study. They qualify all the criteria mentioned above. Though Prophet has not performed well with short time-series, we have considered the model in our study because of its easy applicability and automated forecasting ability (S. Taylor and Letham 2018). NN models have been one of the best performing models according to our literature review; therefore, we have considered both the NN models (MLP and LSTM). SVR models are considered because they have performed well in one of the articles (Cankurt and Subaşi 2016). They can also handle seasonality in the time-series and can generate medium to long term forecasts. Random forests and XGBoost are not considered in the analysis because they usually require more data to perform well (Pedregosa, G. Varoquaux, et al. 2011). In total, we have chosen eight models based on the criteria mentioned for our analysis: Seasonal naive, Holt-Winters, State-space, SARIMA, Prophet, MLP, LSTM and SVR.

4.5 Model descriptions

This section describes all the selected models mathematically. Assume that there is a time-series, Z (where $Z = Z_1, Z_2, \dots Z_t$ for each time period t) with components level (L_t), trend (T_t) and seasonality (S_t). m denotes the seasonal period (for quarterly observations $m = 4$, for monthly observations $m = 12$, etc). Let $\hat{Z}_{t+h}$ represent the forecast value of Z, h-steps ahead of the the observed value at time t.

4.5.1 Seasonal Naive

This model is the most simple one among the selected ones. This model is also the currently employed forecasting model in the case company. The model is of the form: $Z_t = Z_{t-m}$, where the current observation value at time t is equal to the observed value at time $t - m$ and m is the seasonal period. This model serves as a base-model to compare its performance with the complex models and to select those models whose performance is better than seasonal naive.

4.5.2 Holt Winters

The following equations are employed for a multiplicative Holt-Winters model:

$$\begin{aligned}
\text{Forecast equation:} \quad & \hat{Z}_{t+h} = (L_t + hT_t)S_{t-m+h} && (4.6) \\
\text{Level equation:} \quad & L_t = \alpha(Z_t / S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1}) && (4.7) \\
\text{Trend equation:} \quad & T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} && (4.8) \\
\text{Seasonal equation:} \quad & S_t = \gamma \frac{Z_t}{L_{t-1} + T_{t-1}} + (1-\gamma)S_{t-m} && (4.9)
\end{aligned}$$

In the above equations, the values of the smoothing constants, α, β and γ, lie in the range [0,1].The trend equation does not change in the additive model, but the level and seasonal components equation changes in additive model. The seasonal components are summed and subtracted in an additive model instead of being multiplied and divided as in multiplicative model.

The following equations are employed for an additive Holt-Winters model:

$$\begin{aligned}
\text{Forecast equation:} \quad & \hat{Z}_{t+h} = L_t + hT_t + S_{t-m+h} && (4.10) \\
\text{Level equation:} \quad & L_t = \alpha(Z_t - S_{t-m}) + (1-\alpha)(L_{t-1} + T_{t-1}) && (4.11) \\
\text{Trend equation:} \quad & T_t = \beta(L_t - L_{t-1}) + (1-\beta)T_{t-1} && (4.12) \\
\text{Seasonal equation:} \quad & S_t = \gamma(Z_t - L_{t-1} - T_{t-1}) + (1-\gamma)S_{t-m} && (4.13)
\end{aligned}$$

A weighted average between the seasonally adjusted observation $(Z_t - S_{t-m})$, and the non-seasonal forecast $(L_{t-1} + T_{t-1})$ is demonstrated in the level equation 4.11. Similarly, the seasonal equation 4.13 reveals a weighted average between the current seasonal index $(Z_t - L_{t-1} - T_{t-1})$, and the seasonal index of the same season previous year (that is, m time periods ago).

4.5.3 State-space model (ETS)

Rob J Hyndman and Athanasopoulos (2018) demonstrated an alternative representation, *'Component form'* of exponential smoothing equations containing a forecast and smoothing equation.

$$\begin{aligned}
\text{Forecast equation:} \quad & \hat{Z}_{t+1|t} = L_t && (4.14) \\
\text{Smoothing equation:} \quad & L_t = \alpha Z_t + (1-\alpha)L_{t-1} && (4.15)
\end{aligned}$$

where L_t is the smoothed value of the series for time t. The smoothed value for time $t + 1$ is predicted using the smoothing equation 4.15. This is nothing but simple exponential smoothing. Re-arranging the terms in equation 4.15, we get the error correction form:

$$L_t = L_{t-1} + \alpha(Z_t - L_{t-1})L_t = L_{t-1} + \alpha\epsilon_t$$

where $\epsilon_t = Z_t - L_{t-1} = Z_t - \hat{Z}_{t|t-1}$ is the residual (or error) at time t. If ϵ denotes the noise with the distribution $\epsilon\ N(0, \mu^2)$, then the state space equations could be written as:

$$Z_t = L_{t-1} + \epsilon_t \tag{4.16}$$

$$L_t = L_{t-1} + \alpha\epsilon_t \tag{4.17}$$

where equation 4.16 is the *measurement* (or observation) equation which describes the observed data and equation 4.17 is the *state* (or transition) equation which describes the change in the unobserved components or states (level, trend, seasonal) over time (here only change in level is indicated in equation 4.17). Each model consists of a measurement equation that describes the observed data, and some state equations that describe how the unobserved components or states (level, trend, seasonal) change over time and hence these are referred to as state space models. In this study, we will be using state space model with additive error, trend, and seasonality. If our data consists of multiplicative components, they can be transformed to additive components by utilizing logarithm transformation on the data as mentioned in the section 5.2. ETS(AAA) model with log-transformation is considered in this study and it will denoted as Log-ETS(AAA) from here on. It is defined as:

$$\text{Forecast equation:} \quad Z_t = L_{t-1} + T_{t-1} + S_{t-m} + \epsilon_t \tag{4.18}$$

$$\text{Level equation:} \quad L_t = L_{t-1} + T_{t-1} + \alpha\epsilon_t \tag{4.19}$$

$$\text{Trend equation:} \quad T_t = T_{t-1} + \alpha\beta\epsilon_t \tag{4.20}$$

$$\text{Seasonal equation:} \quad S_t = S_{t-m} + \gamma\epsilon_t \tag{4.21}$$

where $\epsilon_t = Z_t - L_{t-1} - T_{t-1} - S_{t-m}$ and α, β and γ, are the smoothing parameters.

4.5.4 SARIMA

The autoregressive (AR(p)) part of the ARIMA model captures the autocorrelation between the present and the past values. Formally, it is written as: $Z_t = \phi_1 Z_{t-1} + \cdots + \phi_p Z_{t-p} + \epsilon_t$ where ϵ_t stands for white noise (ϵ_t belongs to the same distribution with zero average and constant variance), ϕ stands for autoregressive coefficients. The Moving Average (MA(q)) part of the ARIMA model captures the past forecast errors of the model. The corresponding equation is $Z_t = \theta_1\epsilon_{t-1} + \cdots + \theta_q\epsilon_{t-q} + \epsilon_t$ where θ stands for the forecasted error coefficients of the MA(q) model. An important prerequisite for ARIMA models is that the time-series should be stationary (Nau 2020). We can convert a non-stationary series to stationary series by differencing. Formally, the first order of differencing is written as: $Z_t^{'} = Z_t - Z_{t-1}$ and second order differencing is written as $Z_t^{''} = Z_t^{'} - Z_{t-1}^{'}$. For seasonal time-series, we difference the time-series based on the seasonal period m. For a time-series demonstrating seasonal behaviour after m time intervals, then the differencing is done in the following manner: $Z_t^{'} = Z_t - Z_{t-m}$.

Combining the differencing with the autoregression and moving average models, we obtain the non-seasonal ARIMA model. It could be written as:

$$Z_t^{'} = \delta + \sum_{i=1}^{p} \phi_i Z_{t-i}^{'} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \epsilon_t \tag{4.22}$$

In the equation 4.22, $Z_t^{'}$ is the differenced series (it may be differenced more than once), constant δ gives the initial level of the model (similar to intercept in a linear regression) and is calculated using the following equation: $\delta = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p)$ where μ stands for average of the stationary process(Rob J Hyndman and Athanasopoulos 2018; Nau 2020). ϕ_p and θ_q are the parameters of the procedures: AR, with lag length p, and MA, with lag length q.

The constant δ may be omitted if the order of differencing is greater than one (d¿1). If the series is stationary in its original form (d=0), but not with zero average and unit standard deviation, δ is required. Additionally, when the model is devoid of the autoregressive part (AR(p)), it is assumed that the constant is equal to the time series average ($\delta = \mu$) (Nau 2020; Parmezan et al. 2019).

When there is seasonal behaviour in the time-series, we use the seasonal ARIMA, also known as SARIMA, model. It has additional three parameters to capture the seasonal behaviour, which is $(P, D, Q)_m$. D indicates the degree of seasonal difference. The seasonal part of the ARIMA model is written as:

$$Z_t^{''} = \delta + \sum_{i=1}^{P} \Phi_{im} Z_{t-i}^{''} + \sum_{i=1}^{Q} \Theta_{im} \epsilon_{t-im} + \epsilon_t \tag{4.23}$$

In the equation 4.23, $Z_t^{''}$ is the seasonally differenced series, constant δ gives the initial level of the model and is calculated using the following equation: $\delta =$

$\mu(1-\Phi_1-\Phi_2-\cdots-\Phi_p)$ where μ stands for average of the stationary process. The rules for the use of the constant δ and its omission is similar to the ones imposed on the ARIMA structure mentioned above, but considering seasonal difference (D). Φ_P and Θ_Q are the parameters of the procedures: seasonal AR, with lag length P, and seasonal MA, with lag length Q. A SARIMA model of the order $(p, d, q)\times(P, D, Q)_m$ is the sum of the non-seasonal (equation 4.22) and seasonal part (equation 4.23). It is denoted as:

$$Z_t = \delta + \sum_{i=1}^{p} \phi_i Z'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{P} \Phi_{im} Z''_{t-i} + \sum_{i=1}^{Q} \Theta_{im} \epsilon_{t-im} + \epsilon_t \tag{4.24}$$

In the equation 4.24, δ is calculated as: $\delta = \mu(1 - \phi_1 - \phi_2 - \cdots - \phi_p)(1 - \Phi_1 - \Phi_2 - \cdots - \Phi_p)$ and it can omitted when $d + D > 1$. δ is required when $d + D \leq 1$. $\delta = \mu$ is assumed when the model is devoid of autoregressive and seasonal autoregressive parts. Since we are using quarterly data, we substitute $m = 4$ in the equation 4.24 to obtain the final equation that will be used in the analysis.

$$Z_t = \delta + \sum_{i=1}^{p} \phi_i Z'_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} + \sum_{i=1}^{P} \Phi_{4i} Z''_{t-i} + \sum_{i=1}^{Q} \Theta_{4i} \epsilon_{t-4i} + \epsilon_t \tag{4.25}$$

4.5.5 Prophet

In a Prophet model, we use a decomposable time series model with three main model components: trend, seasonality, and holidays (S. Taylor and Letham 2018). It is defined as follows:

$$Z(t) = T(t) + S(t) + H(t) + \epsilon_t \tag{4.26}$$

In the equation 4.26, T(t) captures the non-periodic changes(trend) in the time-series, S(t) captures the periodic changes (weekly and yearly seasonality), H(t) captures the effects of holidays or special events in the time-series, and ϵ_t is the irreducable error term. It is similar to a generalized additive regression model (GAM) with interpretable parameters. The advantages of GAM are that it decomposes easily and accomodates new components as required, such as a new source of seasonality, when recognized, could be added easily to the model. As mentioned earlier in Chpater 3, Prophet implements two models to forecast trend, a saturating growth model and a piece-wise linear model, with the changes in trend captured using the changepoints parameter. The periodic effects (seasonality) is modelled using Fourier Series.

4.5.6 Multilayer Perceptron (MLP)

A perceptron is similar to a single neuron in the human nervous system. The structure of a perceptron is shown in the figure 4.7. A single perceptron consists of *n* data inputs $Y_i \in Y$. Each element of the input is associated with a synaptic weight, w, such that the *i*th element of Y has a synaptic weight w_i associated with it. The weight could be positive or negative based on the significance of the input. The net value is a result of linear combination of weighted inputs with an added bias (where $b \in \mathbb{R}$) (Ismail Fawaz et al. 2019; Parmezan et al. 2019). The resulting net value is sent to an non-linear activation function *f* that determines the output Z of the perceptron. The net value is calculated as shown in the equation 4.27.

$$net = \sum_{i=1}^{n} w_i Y_i + b \tag{4.27}$$

The bias aims to rectify the net value by increasing or decreasing it such that the result of *f(net)* is closest to the expected value. The non-linear activation function *f* is a staircase (or step) function and aids in mapping the non-linear relationship between the inputs and output. Some of the activation functions are *sigmoid, relu* and *tanh*. *Sigmoid* function output range between 0 and 1, *tanh* has negative values and therefore is not considered in the study. Rectified linear unit (*relu*) has linear (identity) for all positive values and is zero for negative values. We have chosen *relu* as the activation function in our study because it was the most popular function in the literature, it provided better results and was computationally more efficient Buxton et al. (2019), Delic (2019), and Parmezan et al. (2019). The output Z could be binary (where $Z \in \{0, 1\}$), or $Z \in \{-1, 1\}$, as well as continuous where $Z \in \mathbb{R}$. A MLP is a combination of more than

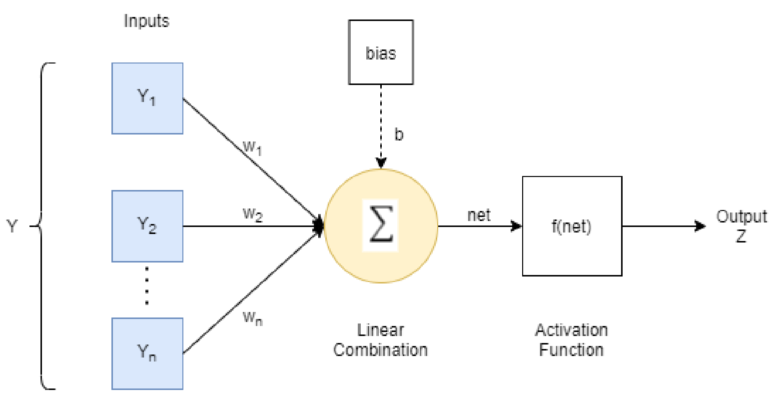

Figure 4.7: Structure of a perceptron

one perceptron. It usually consists of an input layer, an output layer, and hidden layers which are intermediate between the input and output layers. A simple MLP structure with one hidden layer is shown in the figure 4.2. Mathematically, a MLP model with a single hidden layer is defined as shown in the equation 4.28.

$$Z = f(\sum_{j=1}^{k} w_j f(\sum_{i=1}^{n} w_{ij} Y_i + b_{0j}) + b_0) \tag{4.28}$$

where Z is the output, f is the activation function, w_j is the weight assigned to the signal between the hidden layer and the output layer, $w_i j$ is the weight assigned to the signal between the input layer and the hidden layer, Y_i is the input signal and b_0 is the bias added to correct each signal.

4.5.7 Long Short Term Memory (LSTM)

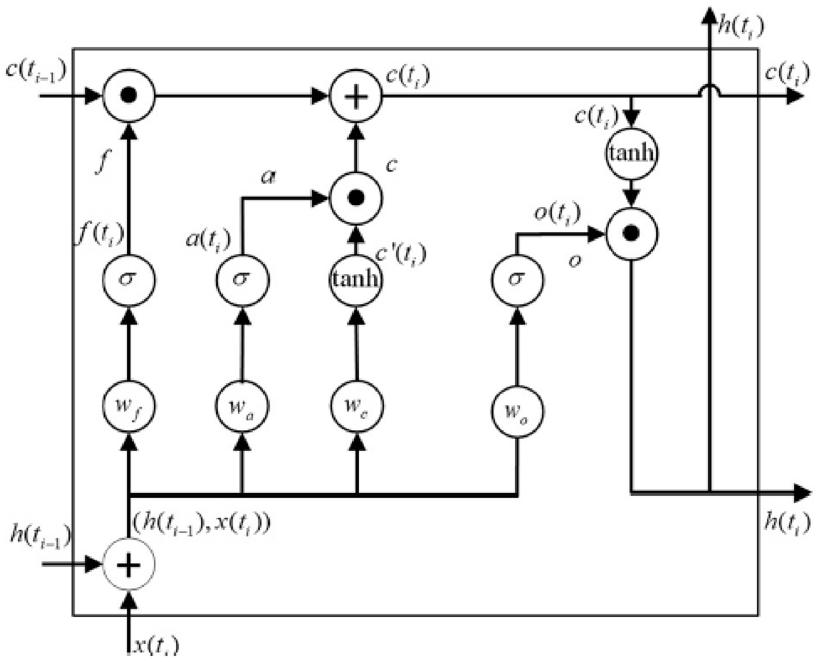

Figure 4.8: Structure of a LSTM cell (Abbasimehr et al. 2020)

LSTM is an extension of recurrent neural network (RNN) and has a strong capability in forecasting time-series due to its ability to store long-range time dependency information (Greff et al. 2017). A LSTM structure comprises an input gate, a forget gate, internal state (cell memory), and an output gate. The structure is shown in the figure 4.3. The notations used in the figure are listed below:

- $x(t_i)$: The input value
- $h(t_{i-1})$ and $h(t_i)$: The output value at time t_{i-1} and t_i
- $c(t_{i-1})$ and $c(t_i)$: Cell states at time t_{i-1} and t_i
- $b = \{b_a, b_f, b_c, b_o\}$ are biases of input gate, output gate, cell state and output gate.
- $W_1 = \{w_a, w_f, w_c, w_o\}$ are the weights of input gate, forget gate, cell state and output gate.
- $W_2 = \{w_{ha}, w_{hf}, w_{hc}, w_{ho}\}$ are the recurrent weights
- $A = \{a(t_i), f(t_i), c(t_i), o(t_i)\}$ are the outputs of input gate, forget gate, cell state and output gate.

Mathematically, the operation of LSTM is defined as follows:

$$a(t_i) = \sigma(w_a x(t_i)) + w_{ha} h(t_{i-1}) + b_a) \tag{4.29}$$

$$f(t_i) = \sigma(w_f x_(t_i)) + w_{hf} h(t_{i-1}) + b_f) \tag{4.30}$$

$$c(t_i) = f_t \times c(t_{i-1}) + a_t \times \tanh(w_c x(t_i)) + w_{hc}(h(t_{i-1}) + b_c) \tag{4.31}$$

$$o(t_i) = \sigma(w_o x(t_i)) + w_{ho} h(t_{i-1}) + b_o) \tag{4.32}$$

$$h(t_i) = o(t_i) \times \tanh(c(t_i)) \tag{4.33}$$

where σ and $\tanh$ are activation functions, and $\times$ indicates point-wise multiplication.

4.5.8 Support Vector Regressor (SVR)

The SVR performs linear or non-linear regression using the concept of transforming the input data into a higher dimensional feature space by linear or non-linear mapping. The figure 4.5 could be referred for formulating the model. The objective function of SVR is to minimize the coefficients rather than the squared error (which is usually the case with ordinary least squares method). The error term is handled in the constraint. The goal of SVR is to find a function $f(x)$ of the line (or hyperplane in higher dimensions) that fit the maximum training data (Awad and Khanna 2015; Chanklan et al. 2018; Xu et al. 2019). The linear function is shown in equation 4.34. The training data is in the supervised learning format (input-output pairs).

$$f(x) = \langle w, x \rangle + b \tag{4.34}$$

where w is the coefficient, x is the predictor (feature), f(x) is the target variable, $w, x \in X$, $b \in \mathbb{R}$ and $\langle ., . \rangle$ indicates dot product in X. The size of the margin from the hyperplane $f(x)$ is indicated by ϵ. SVR tries to find the smallest ϵ deviation from the target value y_i for all the training datapoints. It is calculated as shown in equation 4.35 which is an optimization problem. The optimization problem attempts to find the narrowest tube centred around the hyperplane while minimizing the error (which is the distance between the predicted and desired output).

$$Min\ \frac{1}{2}||w||^2 \quad subject\ to: |y - \langle w, x \rangle| + b \leq \epsilon \tag{4.35}$$

where $||w||$ is the magnitude of the normal vector to the hyperplane that is being approximated. Slack variable (ξ, ξ^*) is introduced to capture the few datapoints outside the margin ϵ (Awad and Khanna 2015). The slack variable should be minimized as much as possible and hence is added to the objective function. The objective function in the equation 4.35 is added with the slack variables and it is defined as:

$$Min\ \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi + \xi^*) \quad Subject\ to: \begin{cases} y - \langle w, x \rangle + b \leq \epsilon + \xi \\ \langle w, x \rangle - y + b \leq \epsilon + \xi^* \end{cases} \tag{4.36}$$

In the equation 4.36, C is a regularization parameter. It determines the choice between the flatness of the hyperplane and the extent of tolerance of deviation from the margin ϵ (Awad and Khanna 2015). C is a tunable parameter where as C increases, the tolerance for data points outside the margin ϵ also increases.

Chapter 5

Model Evaluations

This chapter presents data, the preprocessing steps, model implementation, and the results of modeling. Section 5.1 introduces the time-series data and data preprocessing steps required before modelling. Section 5.2 discusses the data transformation techniques. Section 5.3 presents the grid search method; Section 5.4 discusses the walk-forward validation technique; Section 5.5 presents the performance metrics to support model evaluation; Section 5.6 describes the model implementation and configuration; and finally section 5.7 presents the results of the modeling process.

5.1 Data description and pre-processing

5.1.1 Data description

The data consists of 36 quarterly observations of historical demand data of cage-P to the local customers. It is an univariate series (comprises only of demand variable) which is recorded at the end of each quarter from the year 2011 to year 2019 from the company's database. The data collected is shown in table 5.1. The data is checked for missing values, duplicate values and outliers, and are replaced or removed if found. Missing values and duplicate values are checked using functions available in Python. Presence of outliers are checked using box-plots. We did not find any of the above mentioned irregularities in our collected data. The data is split into training and testing data for cross-validation. The usual thumb rule of 80-20% random split is not used here to preserve the temporal information.

	Year	Q1	Q2	Q3	Q4
Train data	**2011**	24	37	30	15
	2012	26	36	22	14
	2013	34	42	27	19
	2014	35	40	30	15
	2015	38	45	25	18
	2016	39	43	23	16
	2017	47	48	30	18
	2018	55	51	22	24
Test data	**2019**	61	63	32	25

Table 5.1: Demand data collected on a quarterly basis

Statistical measures	Number of cages
Count	36
Mean	32
Standard deviation	13
Min	14
Median	30
Max	63

Table 5.2: Summary statistics of demand data

The years 2011-2018 (consisting of 32 quarterly observations) serve as the training set and the year 2019 (consisting of 4 quarterly observations) serves as the testing set. Multi-step forecasting is used in this study where multiple time-steps must be predicted in contrast to one-step forecasting where only one time-step is predicted. The summary statistics of the data is shown in table 5.2.

The training set of the time-series data is decomposed into three main components, trend, seasonal, and residuals (or random variation). The decomposition is shown in the figure 5.1 and it is evident from the figure that both trend and seasonal components are present. We can notice that trend begins with an upward trend which quickly reverses to a downtrend; then the trend stays flat for a while which is followed by a rapid uptrend from mid-2016 to the peak in the beginning of 2018, and it starts decreasing again. These sudden shifts in trend make forecasting task difficult. Data transformations such as the Box-Cox transformation or logarithm transformation could be used to stabilize the variance. The seasonal behaviour is also evident in the time-series data with the pattern showing consistent peak in the second quarter and then decreases in the consecutive quarters. The residuals are remainder that is left after subtracting the trend and seasonal components from the data. Ideally, it should be independent and identically distributed (that is, should be uncorrelated). As we can see from the plot, there is no visible pattern in the residuals which leads us to conclude that the residuals are uncorrelated. Ljung Box statistic test is also performed to check if the values are correlated. Ljung Box test is a statistical test where it tests the null hypothesis that the data is independently distributed (Rob J Hyndman and Athanasopoulos 2018). We obtained a p-value of 0.161 and since it was greater than 0.05(significant value), we concluded that the residuals are uncorrelated. If the residuals are correlated, then data transformations are necessary to stabilize the variance or exogenous variables might be required to completely explain the time-series.

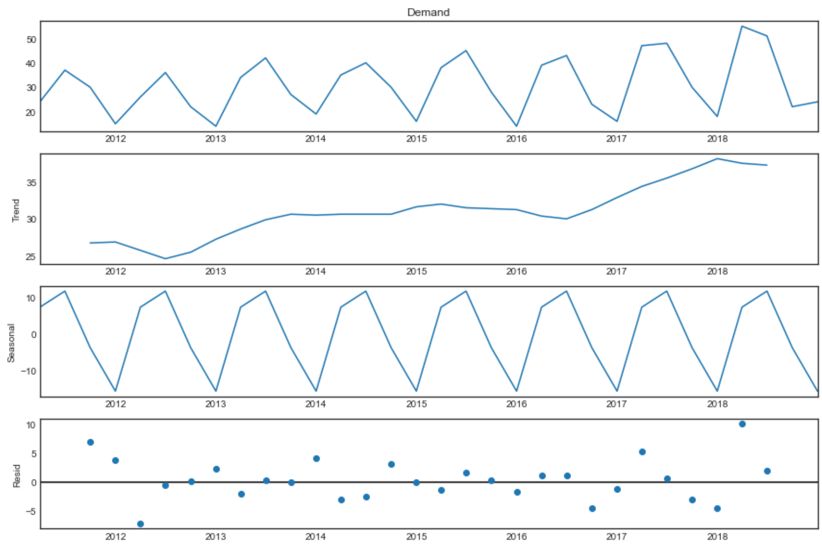

Figure 5.1: Decomposition of time-series data

5.1.2 Stationarity

Apart from checking the data for irregularities and splitting the data into train-test dataset, another important data pre-processing step is checking the data for stationarity. A stationary series is defined as a series with constant mean, constant variance and constant covariance (uncorrelated). Time-series with trend and seasonality components are generally non-stationary. We require stationary time series for especially SARIMA model. We have also used stationary time-series for neural network models in this study. We verify stationarity by using various plots (mainly ACF & PACF plots) and statistical hypothesis tests such as the Augmented Dickey-Fuller Test.

ACF and PACF

The autocorrelation function (ACF) and partial autocorrelation function (PACF) plots are used to inspect the stationarity of the time-series. ACF plot shows the values of autocorrelation of the current value with its lagged values while PACF summarizes the relationship of a particular observation in the time-series with the observations in the previous time-step where the relationships of the intermediary observations are removed (Brownlee 2017). In the plots, if autocorrelations for small lags are large and positive, presence of trend is expected. Seasonal behaviour is expected if the autocorrelations are larger for the seasonal lags (such as a spike at lag 4 and multiples of 4 for *quarterly* time-series) than for the other lags. Lag-0 usually has a high correlation value because the value is correlated to itself and hence it is generally ignored in the analysis. The ACF and PACF plot of the training set of demand data is shown in the figure 5.2. Futhermore, the ACF and PACF plots using the logarithmic and Box-cox transformations on the training set of demand data are shown in the Appendix-A. Visual inspections of these plots help in the identification of required data transformation and the parameters for the ARIMA model. The blue band in the plots indicate the confidence interval of no correlation between the lags, and the y-axis and x-axis are the correlation values and the lags respectively.

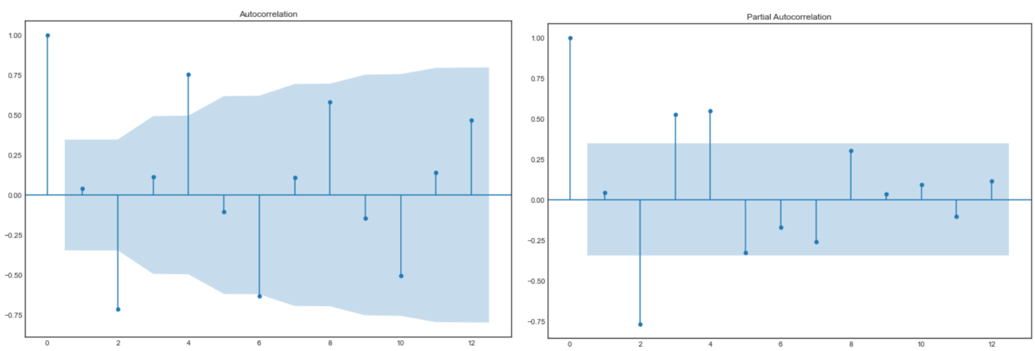

Figure 5.2: ACF and PACF plots of actual training data

The ACF plots of the logarithm and Box-Cox transformed series were expected to demonstrate stabilized variance. This could be examined by comparing the ACF plots of the transformed series with the plot of the actual series and evaluating the reduction in height of peaks and depth of troughs. The plots are shown in Appendix-A. We noticed that there was not a significant reduction in variance on applying both the transformations. Similarly, ACF plots of the transformed series also appeared analogous to the ACF plot of the actual series. Nau (2020) suggested the following rules to identify the order of differencing:

1. A higher order of differencing is required if the series shows positive autocorrelations out to a higher number of lags in the ACF plot.
2. The series does not need a higher order of differencing if the series shows zero or negative autocorrelation at lag-1, or small and patternless autocorrelations. The series may be *overdifferenced*, if the autocorrelation at lag-1 is -0.5 or more negative. Overdifferenced series leads to inaccurate model.
3. The order of differencing at which the standard deviation is the lowest is often the *optimal order of differencing*.

From the figure 5.2, we can notice that a higher order of differencing is unnecessary since the positive autocorrelations dies out after lag-1 and then is significant only at lag-4. Hence, we began with first order of differencing. First order differencing is applied on the training data and the plots are shown in the figure 5.3. The magnitude of negative autocorrelation at lag-1 is below 0.5 which shows that the series was not overdifferenced. Similar to the ACF and PACF plots of the actual data, we see spikes at lag-2 and lag-4 in the ACF plot and at lags 2 & 3 in PACF plot (higher lags are ignored since we aim to create a parsimonious model).

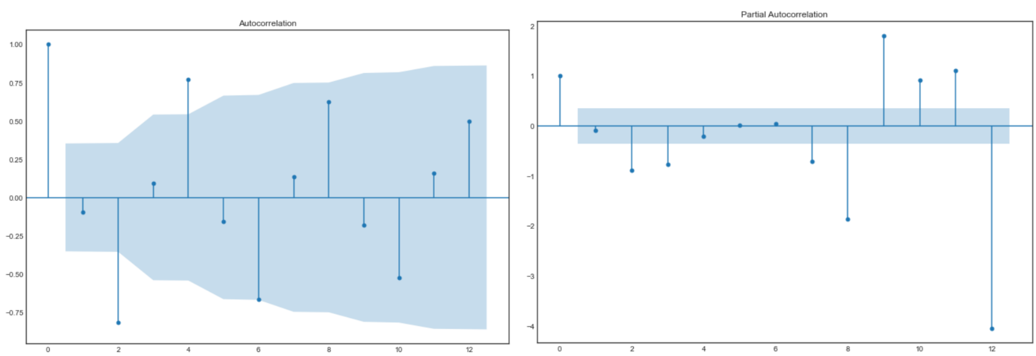

Figure 5.3: ACF and PACF plots of first-order differenced actual training data

Since the series shows seasonal behaviour, we also try to apply seasonal differencing as suggested by Rob J Hyndman and Athanasopoulos (2018) and Nau (2020). The ACF and PACF plots of the seasonal differenced series is shown in the figure 5.4. We noticed that there were no significant spikes in the smaller lags in both the plots.

	Actual	log	Box-Cox	Actual, d=1	log, d=1	Box-Cox, d=1	Actual, D=1	log, D=1	Box-Cox, D=1
Cage-P	0.956	0.905	0.876	**0.001**	**0.002**	**0.001**	**0.001**	**0.0005**	**0.0007**

Table 5.3: P-values of Augmented Dickey-Fuller test for the time-series

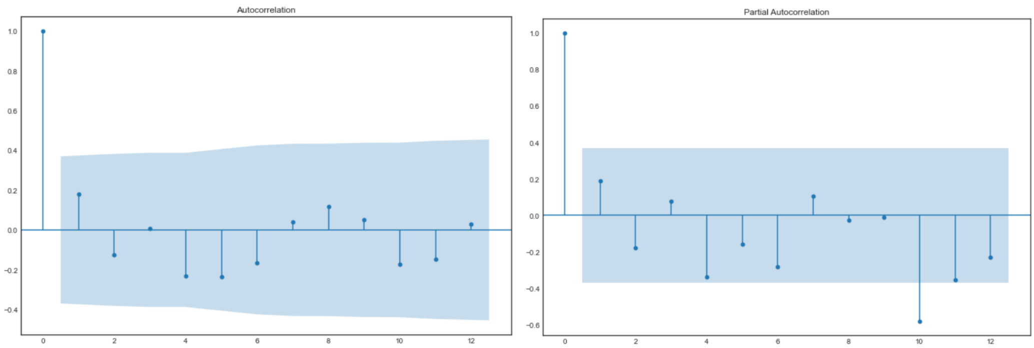

Figure 5.4: ACF and PACF plots of seasonally differenced actual training data

Augmented Dickey-Fuller Test

The Augmented Dickey-Fuller Test is a statistical hypothesis test used to objectively determine whether the time-series is stationary or not. It is a popular statistical test in the field of data science. It uses an AR model and differs the lag value to optimize the information criterion by minimizing it. The information criterion used in this study is Corrected Akaike's Information Criteria (AICc) since it provides stronger penalty than the other popular information criteria (AIC and BIC) on smaller sample sizes. It is defined as:

$$AICc = AIC + 2(\frac{(k+2)(k+3)}{(T-k-3)}) \tag{5.1}$$

where T is the number of observations, k is the number of predictors and AIC is calculated as: $AIC = Tlog(\frac{\sum_{t=1}^{T} \epsilon_t^2}{T}) + 2(k+2)$. Information criteria are mainly useful for simple and parsimonious model selection (Stoica and Selen 2004). The null hypothesis is that the time-series is not stationary. On obtaining a p-value below the threshold (that is, the significant value which is usually 0.05 or 5%), the null hypothesis can be rejected while with a p-value above the threshold, the null hypothesis cannot be rejected. The training data is usually subjected to the statistical test to check for stationarity. The p-values obtained from the test are shown in the table 5.3. The bold-values indicate that the series is stationary on differentiation.

Rob J Hyndman and Athanasopoulos (2018) stated that 'When both seasonal and first differences are applied, it makes no difference which is done first—the result will be the same; however, if the data has a strong seasonal pattern, we recommend that seasonal differencing be done first'. The reason being that the resulting series might be stationary and further first order differencing will not be required. But if first order differencing is done first, then seasonality will still be present.

5.2 Data transformation techniques

The data has to be transformed into a specific format before being used in modeling. The time-series data might require some transformations before being used as an input to the traditional models and the advanced forecasting models require the data to be converted to a supervised learning format. The purpose of applying transformations to the time-series data was to either make the patterns more consistent across the whole data-set or to remove the known sources of variation simplifying the patterns (Rob J Hyndman and Athanasopoulos 2018). Accurate forecasts are usually obtained from a data-set comprising of simple patterns.

Transformations such as logarithmic or power transforms are used when the data shows variance that increase or decrease about the level of the series. These transformations aids in stabilizing the series (Akpinar and Yumusak 2016; Claveria and Torra 2014; Papacharalampous et al. 2018). A logarithm or power transform is also useful in converting multiplicative models to additive models. For example: If $Z_t = T_t \times S_t \times \epsilon_t$ is a time-series with multiplicative components (trend, seasonal and residual components), with logarithm transformation applied to the data we can use an additive model. That is, $Z_t = T_t \times S_t \times \epsilon_t$ is equivalent to $log(Z_t) = log(T_t) + log(S_t) + log(\epsilon_t)$. A family of Box-Cox transformations, which encompasses both logarithm and power transformations serves as handy tool for performing transformation and it depends on a parameter λ. Formally, it is written as:

$$\omega_t = \begin{cases} log(Z_t) & \text{if } \lambda = 0; \\ (Z_t^{\lambda} - 1)/\lambda & \text{otherwise.} \end{cases} \tag{5.2}$$

Some of the traditional models, especially SARIMA model, require a stationary series.A series with trend or seasonality is considered as a non-stationary series. A non-stationary series can be converted to a stationary series by taking a difference of the consecutive observations. This is known as *differencing* (Papastefanopoulos et al. 2020; Parmezan et al. 2019; Shih and Rajendran 2019). For seasonal time-series, we difference the time-series based on the seasonal period. While the variance of the time series could also be stablized using transformations such as logarithms and Box-Cox (Rob J Hyndman and Athanasopoulos 2018), differencing stabilises the mean of a time series by eliminating the variations in the level of a time series. Multiple statistical tests are also available to test the stationarity of the time series such as the Augmented Dickey-Fuller (ADF) test described in the section 5.1.2.

Sliding window method is usually used to convert a time-series data into supervised learning problem (Brownlee 2017). A supervised learning problem is a subset of machine-learning which consists of labelled data (input) and desired output and an algorithm is employed to learn the mapping from the labelled data (input) to the desired output. In sliding window method, the observed values at the previous time-steps are used as input variables (or labelled data) and the next time-step is used as output variable (Brownlee 2017). Sliding window method is also know as *lag method*. An example of sliding window method is shown in figure 5.5. As we can see in the figure, the order between the observations is preserved, and the top row and the last row is deleted in the last table. The reason is that there is no value that can be used as a labelled data (X) in the top row and there is no value that serves as a desired outcome (Z) in the last row. The width of window can be increased if desired and the width of the window could be determined from the partial autocorrelation graphs of the time series (Brownlee 2017).

Time	Sales
Q1_16	50
Q2_16	59
Q3_16	35
Q4_16	29
Q1_17	55
Q2_17	54
Q3_17	37
Q4_17	32

→

X	Z
?	50
50	59
59	35
35	29
29	55
55	54
54	37
37	32
32	?

→

X	Z
50	59
59	35
35	29
29	55
55	54
54	37
37	32

Figure 5.5: An example of sliding window method (adapted from (Brownlee 2017))

5.3 Grid searching

Grid searching is an iterative process in which optimal parameters for a model are selected by iterating through every parameter combination, storing a model for each combination and scoring each model. The model with best score is considered to have optimal parameters and generally optimal parameters help to improve model performance (Abbasimehr et al. 2020; Brownlee 2018; Papacharalampous et al. 2018; Parmezan et al. 2019). Grid-searching is computationally expensive when used on a large dataset and it takes a while to compute the best parameters. But, since the dataset used in our study is small, it is possible to use grid-search to find the optimal parameters (Brownlee 2018). In this study, grid-search is used to find optimal parameters for exponential smoothing models and to find hyper-parameters for all the chosen advanced models expect Prophet. The range of hyper-parameters was determined from the literature and by hand-tuning. The range of each parameter used in all the chosen models are discussed under the respective model description.

5.4 Walk forward Validation

Evaluation of time-series forecasting models can be performed by using walk-forward validation on a test set (Brownlee 2017; Rob J Hyndman and Athanasopoulos 2018). Walk-forward validation is a method where the model makes a forecast, one at time for a time step, for each observation in the test data and the true observation is added to the test dataset. The true observation is made available to the model as a part of the input for making the prediction on the next time step. The observations could be refit in simpler models before making the subsequent prediction, but cannot be refitted in complex models, such as neural networks, due to the greater computational cost. This method is generally used with machine learning models. The method is shown in the figure 5.6. In our study, year 2011 - 2017 is used as a first training set and year 2018 is used as the first test set and the prediction is made for each quarter of 2018. Once the first quarter has been predicted, the actual observed value in the first quarter is added to the test data and it is used as a part of the input to predict the second quarter in 2018. This process continues until all the quarters in the subsequent years have been predicted. However, only the predictions made in the year 2019 are considered in the results.

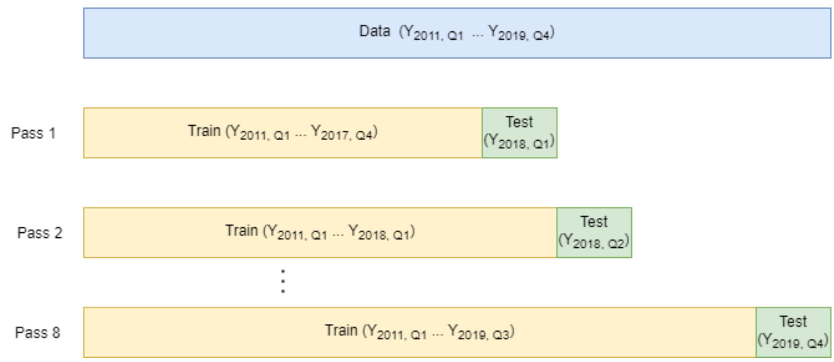

Figure 5.6: Walk forward validation

5.5 Performance metrics

This section discusses various performance metrics that will be used in the later chapters to measure the performance of the forecasting methods. The forecast accuracy forms a vital component in measuring the performance of a model. The forecast error is calculated as the difference between the actual observed value at time t and the predicted value at time t. There has been a variety of metrics that has been used in the literature for the evaluation of performance of forecasting methods (Rob J. Hyndman and Anne B. Koehler 2006). Each metric has its own advantages and disadvantages in comparing the results of the models and its suitability depends on the type of data. From the table 4.3, we can see that Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) are the most frequently used metrics. Hence, we have selected the RMSE as the performance measurement metric due to their explainability, relevance, and popularity, to measure the performance of both the traditional and advanced forecasting methods. MAPE is scale-independent and is useful when two different datasets with different scales are used in the analysis (Rob J. Hyndman and Anne B. Koehler 2006). Since we will use only one dataset for all the models, we will not include MAPE in the study. In addition to RMSE, we have also included coefficient of determination (R^2), which measures the goodness of fit of the model to the data-set. The following sub-sections discusses the performance metrics in details.

5.5.1 Root Mean Squared Error (RMSE)

RMSE is a scale-dependent accuracy measure where the scale of the measure depends on the scale of the data (Rob J. Hyndman and Anne B. Koehler 2006). Therefore, it is useful when comparing different methods are used on the same dataset and it cannot be used when comparing datasets that have different scales. It is also sensitive to extreme values in a distribution and it penalizes the larger errors compared to MAE (Rob J. Hyndman and Anne B. Koehler 2006). Moreover, RMSE is related to the least squares based cost function used in most of the models (Rob J Hyndman and Athanasopoulos 2018; Swalin 2018), but RMSE is more sensitive to outliers compared to MAE (Rob J. Hyndman and Anne B. Koehler 2006). These reasons compelled us to use RMSE as a performance metric in the model evaluation. It is calculated as the root of the average of squared differences between predicted value and observed value. If Z_t is the actual observed value at time t and $\hat{Z}_t$ is the predicted value at time t, then RMSE is defined as:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(Z_t - \hat{Z}_t)^2}{n}} \tag{5.3}$$

Lower the RMSE, the better is the accuracy and thus the performance of the model.

5.5.2 Coefficient of determination (R^2)

The coefficient of determination (R^2) demonstrates the goodness of fit for a model, that is, it explains how well the independent variable explains the variability in the dependent variable in a regression model (Swalin 2018). The value of R^2 ranges from 0 to 1, where 1 indicates a perfectly fit model. Occasionally, models get negative R^2 values. This means that the model is not following the trend in the data and is performing worse than a horizontal line. Formally, R^2 is calculated as follows:

$$R^2 = 1 - \frac{\sum_{t=1}^{n}(Z_t - \hat{Z}_t)^2}{\sum_{t=1}^{n}(Z_t - \bar{Z}_t)^2} \tag{5.4}$$

where Z_t is the observed value at time t, $\hat{Z}_t$ is the predicted value at time t and $\bar{Z}_t$ is the mean of observed values at time t which is given by $\bar{Z}_t = \sum_{t=1}^{n}(Z_t/n)$. R^2 is scale-independent and it is sensitive to the variance present in the observations.

5.6 Model implementation and configuration

In this section, we discuss the model implementation in Python environment and also the configuration of each model in terms of optimal parameters. In general, the modeling procedure follows the steps as shown in the schematic diagram 5.7. The comparison of the models is discussed in Chapter 6.

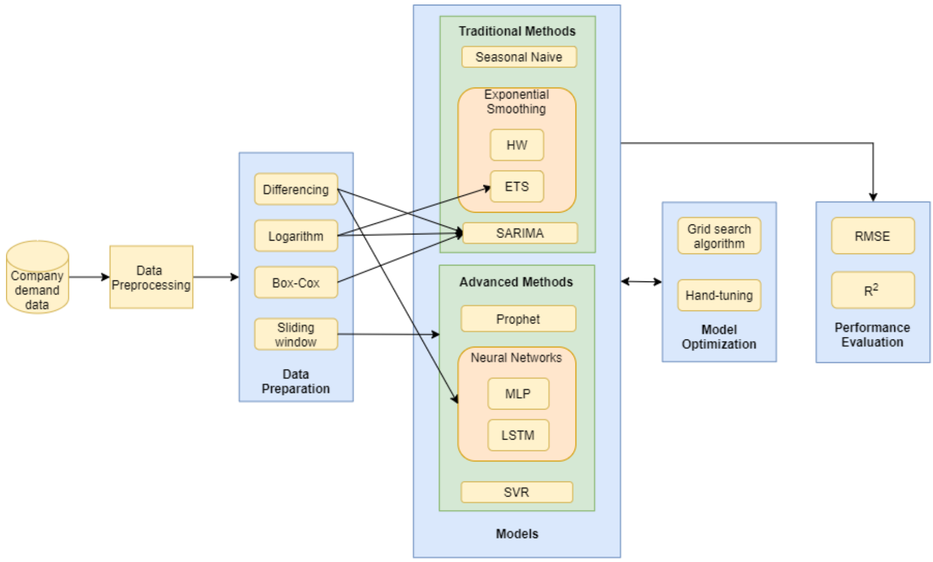

Figure 5.7: Schematic diagram of modeling approach

5.6.1 Seasonal Naive

The model was implemented in Python by writing a custom function because of the unavailability of a package. Since the model does not contain any parameters, parameter selection using grid search was not necessary to obtain an optimized model.

5.6.2 Holt Winters

The model was implemented using the statsmodels.tsa.holtwinters package (Seabold and Perktold 2010) in Python for generating predictions. Grid searching was used to select the best parameters for each model. The parameters used in the grid-search for Holt-Winters model are:

- Trend: The first parameter was the nature of trend component with two options, additive and multiplicative.
- Seasonal: The second parameter was the nature of seasonal component with two options, additive and multiplicative.
- Damped: The third parameter was to indicate the presence of damped trend. It was a binary parameter with values *True* and *False*.
- Box-Cox: The last parameter was to implement the Box-Cox transformation on the time-series. It had three values, they were *True,False* and *log*.

The grid-search resulted in a total of 24 models with different combinations used in each model. AICc was chosen as the information criterion in the model selection due its superior performance with smaller sample sizes. The optimization objective of all the models was to minimize the AICc on the training data. The model that resulted in the least AICc value on the train data, and RMSE value on the train data and test data was chosen. The best performing model was optimized automatically by maximizing the log-likelihood and had the following parameters:

- Trend: Additive
- Seasonal: Multiplicative
- Damped: False
- Box-Cox: False

The multiplicative Holt-Winters model (refer to equation 4.6) was used to generate forecasts(h=4 in the equation 4.6). The values of the smoothing paramaters in the best performing model were: $\alpha = 0.37, \beta = 0.05$ and $\gamma = 0.63$. It shows that the initial observations has more influence than the latest observations on the level and trend variables of the forecast equation due to the low value of α and β. While the high value of γ indicates that the seasonality has been learnt from the recent observations.

5.6.3 State Space model (ETS)

The model was implemented using the statsmodels.tsa.statespace in Python environment. The ETS model also had similar parameters as those of Holt-Winters model. Since, the best model is Holt-Winters model had multiplicative seasonality, it was evident that a multiplicative model was used in modeling the train data.

The statsmodels.tsa.statespace package does not have multiplicative seasonality to date; therefore, it is necessary to perform data transformation on the train data before modeling. We have applied logarithm transformation on the train data to convert the components to additive form so that additive model could be used for modeling. The only difference in the parameters of Log-ETS(AAA) model was the seasonal parameter value. In this model, we had to indicate the seasonal periods rather than the type (that is, additive or multiplicative). The value of seasonal periods was 4 due to the quarterly observations used in the study. The additive model in equation 4.18 was used for generating forecasts. The values of smoothing parameters of the best performing model were: $\alpha = 0.0001, \beta = 0.0001$ and $\gamma = 0.9550$. The values of α and β was close to zero and high value of γ indicates that the seasonality had higher influence on the final forecast equation.

5.6.4 SARIMA

The SARIMA model was implemented in Python using the statsmodels.tsa.sarimax package (Seabold and Perktold 2010). We use the *auto_arima* method in the pmdarima library (G Smith 2020) to perform automatic model selection. The *auto_arima* method performs grid search to return the optimal values for the parameters (p,d,q,P,D,Q) with the optimization objective to minimize AICc. The best SARIMA model according to the *auto_arima* method was of the order $(0,0,0)\times(0,1,0)_4$.

The parameters for the SARIMA model can also be found manually and this could be used to validate the result obtained from the automatic model selection method. The values of the parameters can be found from the ACF and PACF plots and the rules for determining the orders of the model are described by Nau (2020). Since the series is non-stationary, differencing is required to make it stationary. As discussed in the section 5.1.2, seasonal differencing is preferred over first-order differencing. After performing the seasonal differencing, the ACF and PACF plots (refer figure 5.4 were analysed and it was found that there were no spikes found in any of the smaller lags (preferably lags 1-4). Hence, SARIMA model of the order $(0,0,0) \times (0,1,0)_4$ was found to be the best model and this was similar to the result obtained from the *auto_arima* method. The constant δ was included in the final model (because $d + D \leq 1$) with a value of *1.643*.

Though the visual inspection of the ACF and PACF plots in Section 5.1.2 did not reveal any necessary data transformations, we have considered logarithm transformed series and Box-Cox transformed series to examine if they can enhance the forecasting performance. Again *auto_arima* method was used to find the best orders for the parameters of both the models. The automatic model selection method resulted in the best SARIMA model of the order $(0,0,0) \times (1,1,0)_4$ for both the transformed series. The constant δ was included in the final models (because $d + D \leq 1$) with a value of *0.075* and *0.304* for the log-transformed series and Box-Cox transformed series models. The value of the coefficient of the seasonal autoregressive part (Φ_P) was *-0.420* and *-0.410* for the log-transformed series and Box-Cox transformed series models.

These obtained order of the parameters were utilized in building the final SARIMA model for each of the series (that is with actual series, logarithm transformed series and Box-Cox transformed series).

5.6.5 Prophet

The Prophet model was implemented in Python using the fbprophet library developed by Facebook. The data had to be converted to a specific format before modeling. The data was converted into a series with two columns *ds* and *y*, where the *ds* column comprises all the dates and the *y* comprises the demand data. Grid searching was not performed to find the optimal parameters because the parameters were intuitive. The values of the parameters were mostly default and some of them were found by hand-tuning. The most important parameters mentioned for the model were:

- growth - This parameter identifies the type of trend model to use while modelling. By default, it is 'linear' and the default value was used in our model since our data did not show any signs of saturation (refer the graph in figure 5.1).
- yearly seasonality - This parameter indicates the presence of periodic changes on an yearly basis.
- number of changepoints - This parameter indicates the number of points in the time-series at which there are sudden changes in trend. We set the value to be 10 which was obtained by hand-tuning.
- seasonality - This parameter indicates the type of seasonality to be used in the model. We chose the seasonality to be of *multiplicative* because the seasonal variation was increasing with the level of the time-series.

Since the model is considered to be an automatic forecasting tool, most of the initial data preprocessing (such as removing outliers, duplicate values or missing values) could be skipped and a raw time-series (without any data pre-processing steps performed on it) could be used.

5.6.6 Multilayer Perceptron (MLP)

The MLP model was implemented in Python using the Keras library (Chollet et al. 2018). The data was converted to supervised learning format using the sliding window method (as mentioned in Chapter 3) with a window width of 4 (that is 4 inputs are mapped to a single output). The grid search was performed to find the optimal hyperparameters for the model. The list of important hyperparameters that was used in the grid search were:

- n_inputs - The number of lag observations to be used as an input to the model. The value is synonymous to the window width in the sliding window.
- n_epochs - This parameter indicates the number of times to expose the model to the whole training data. We selected two values, 20 and 50. Bigger numbers were not selected to avoid overfitting of the models due to small train data size.
- n_nodes - The number of nodes to be used in the hidden layer. The values selected were 20 and 30. Bigger numbers were not selected to avoid overfitting of the models due to small train data size.
- n_batch - The number of samples considered within an epoch after which the weights are updated. We selected three values for this parameter, 1,4 and 16. Bigger numbers were not selected to avoid overfitting of the models due to small train data size.
- n_diff - The number of differencing required to stationarize the series. We selected three values, 0,1, and 4 (equivalent to seasonal differencing).

There were a total of 24 model configurations and the best model (with the least RMSE) had the following parameters: n_inputs=4, n_nodes=30, n_epochs=50, n_batch=4, n_diff=4. The best model preferred seasonal differencing over first-order differencing, and is in accordance with the rule discussed in the end of section 5.1.2. Default values were used for dropout rate (helps in model generalization) and learning rate (adjusts the extent of change to the model weights) parameters. These parameters were used in model fitting. The optimizer used for the model fitting was *adam* and the loss function to be optimized was *mean squared error*. Adaptive moment estimation (adam), an optimization algorithm for the weights, was used as the optimizer in this study because it is computationally efficient and can handle sparse gradients. Walk forward validation was used to generate forecasts for the four quarters in the test data.

5.6.7 Long Short Term Memory (LSTM)

The LSTM model was implemented in Python using Keras library (Chollet et al. 2018). The data was converted to supervised learning format using the *sliding window* method (as mentioned in Chapter 3) with a window width of 4 (that is 4 inputs are mapped to a single output). The grid search was performed to find the optimal hyperparameters for the model. The list of important hyperparameters and its ranges were similar to that of MLP model.

There were a total of 24 model configurations and the best model (with the least RMSE) had the following parameters: n_inputs=4, n_nodes=30, n_epochs=50, n_batch=1, n_diff=4. Similar to the best model in MLP, the best model of LSTM also preferred seasonal differencing over first-order differencing, and is in accordance with the rule discussed in the end of section 5.1.2. Default values were used for dropout rate and learning rate parameters. These parameters were used in model fitting. Similar to MLP, the optimizer used for the model fitting was *adam* and the loss function to be optimized was *mean squared error*. Walk forward validation was used to generate forecasts for the four quarters in the test data.

5.6.8 Support Vector Regressor (SVR)

The SVR model was implemented in Python using the sklearn library (Pedregosa, G. Varoquaux, et al. 2011). The data was converted to supervised learning format using the sliding window method with a window width of 4. The grid search was performed to find the optimal hyperparameters for the model. The list of hyperparameters that were used in the model were:

- C - C is a regularization term which controls the tolerance for data points outside the margin. It adds penalty to the mis-classified datapoint. The value of C is usually in the range of 0.1 to 100.
- kernel - The kernel aids in finding a hyperplane in the higher dimensional space without increasing the computational cost. The most popular kernals used in practice are radial basis function (RBF), linear and polynomial.

The optimal hyperparameters that were chosen from the grid search were C: 100

and kernal: 'linear'. Walk forward validation was used to generate the forecasts for the four quarters in the year 2019.

5.7 Results

This section presents the results obtained from each of the models in the form of graphs, and the performance metrics obtained.

5.7.1 Seasonal Naive

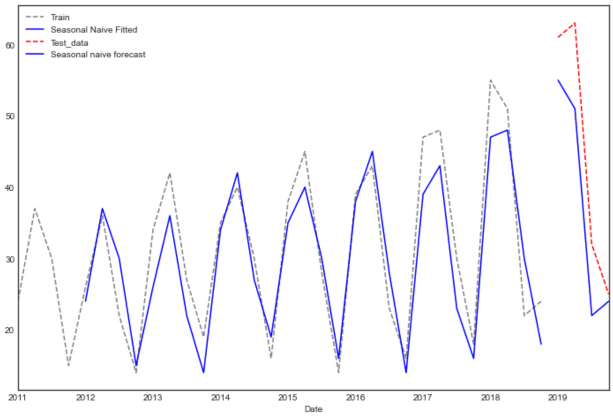

Figure 5.8: Seasonal Naive forecasts

The result of the prediction is shown in the figure 5.8. In the figure 5.8, the blue-line indicates the forecast values, grey-line indicates the train-data and the red-line indicates the test-data. The fitted values on the train data indicate that were obtained from the same quarter of the previous year. We can notice that the second and third quarters have greater difference between the predicted values and the actual values (test-data) compared to the first and fourth quarters. The RMSE score and R^2 values were evaluated to capture the performance of the model and we achieved the values as **8.4** and **0.76** respectively. The residuals were also examined and they were uncorrelated and normally distributed. Various residual plots (refer Appendix A.4) such as ACF plot and density plot were also inspected to verify the aforementioned residual results.

5.7.2 Holt Winters

The results of the prediction using multiplicative Holt-Winters model is shown in figure 5.9. From the graphs, we noticed that Holt-Winters model generated

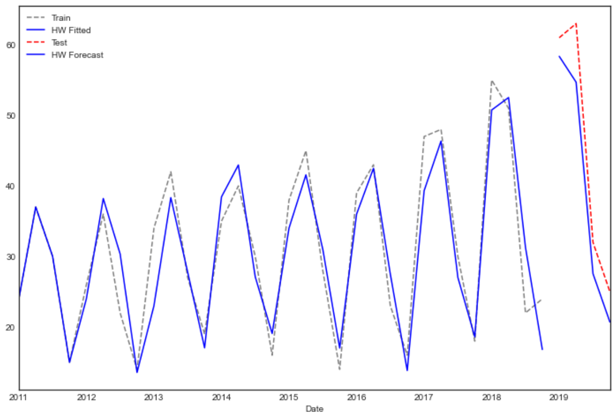

Figure 5.9: Holt-Winters forecasts

forecasts that were under-estimated for all the quarters. The fitted values, indicated as blue-lines, demonstrate correct predictions in some of the quarters in the train data. The performance metrics were calculated and the obtained values for RMSE and R^2 were **5.3** and **0.9** respectively. The residuals were also examined and they were uncorrelated and normally distributed. Various residual plots (refer Appendix A.4) such as ACF plot and density plot were also inspected to verify the aforementioned residual results.

5.7.3 State Space model (ETS)

The results of the prediction of state-space model, Log-ETS(AAA), is shown in figure 5.10. The fitted values demonstrate that the model struggled to fit on the initial quarters in the train data but later on had a good fit. The performance metrics were calculated and the obtained values of RMSE and R^2 were **5.5** and **0.89** respectively. As we can see from the performance metrics, Holt-Winters performed marginally better. Generally, state-space models perform better than the simple exponential smoothing models; but in this study, it was the other way around. The residuals were also examined and they were uncorrelated and normally distributed. Various residual plots (refer Appendix A.4) such as ACF plot and density plot were also inspected to verify the aforementioned residual results.

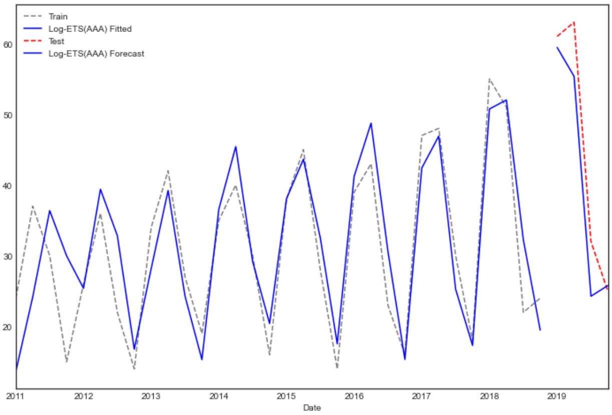

Figure 5.10: Forecasts from ETS(AAA) with log-transformed data

5.7.4 SARIMA

The results of the predictions using actual series is shown in figure 5.11, using the logarithm transformed series is shown in figure 5.12 and the Box-Cox transformed series is shown in figure 5.13.

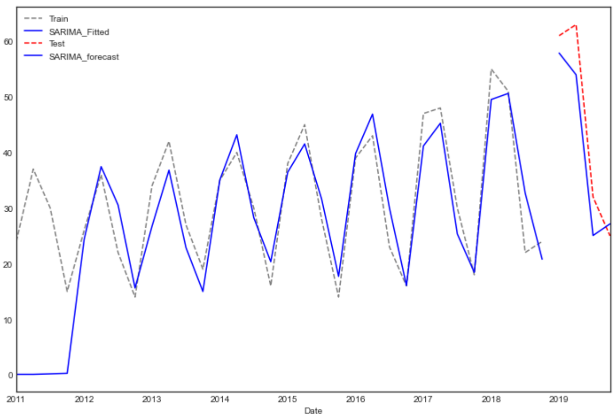

Figure 5.11: Forecasts from SARIMA model with actual series

Additionally, the residuals were also examined and they were uncorrelated and normally distributed. Various residual plots (refer Appendix A.4) such as correlogram, Q-Q plot and histogram were inspected to verify the aforementioned residual results. From the figures, we could say that the models with transformed series performed better than the model with actual series. The performance metrics also reproduced similar results. The actual series model had a RMSE value

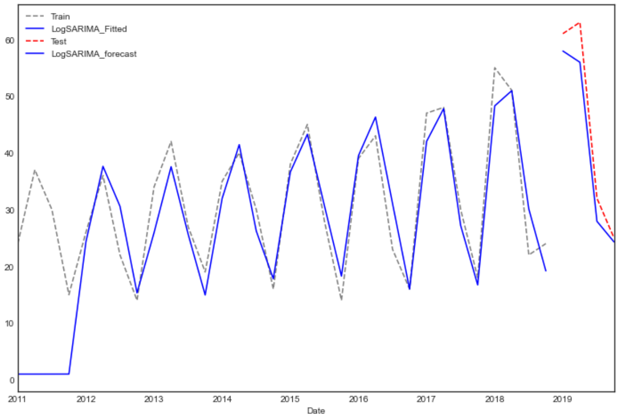

Figure 5.12: Forecasts from SARIMA with log-transformed series

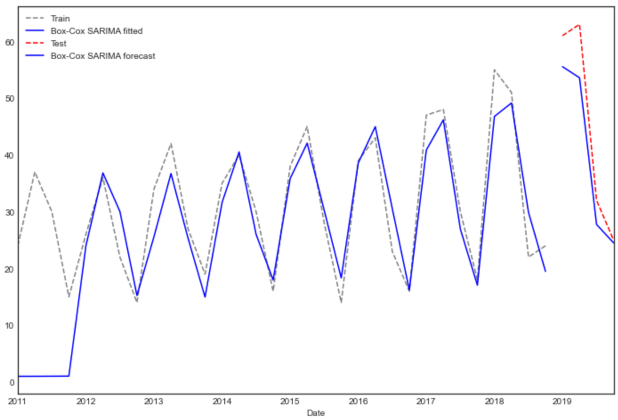

Figure 5.13: Forecasts from SARIMA with Box-Cox transformed series

of **6.0** and R^2 value of **0.87**, while the log-transformed series model had a RMSE value of **4.4** and R^2 value of **0.93**, and the Box-Cox transformed series model had a RMSE value of **5.8** and R^2 value of **0.88**. The log-transformed series model performed the best out of the three models.

5.7.5 Prophet

The results of the predictions using the Prophet model are shown in the figure 5.14. From the graph, we noticed that the model over-estimates the value for the third quarter and under estimates for the remaining quarters. Also, the fitted values were absent in the plot because of the model's unexplainability which is due

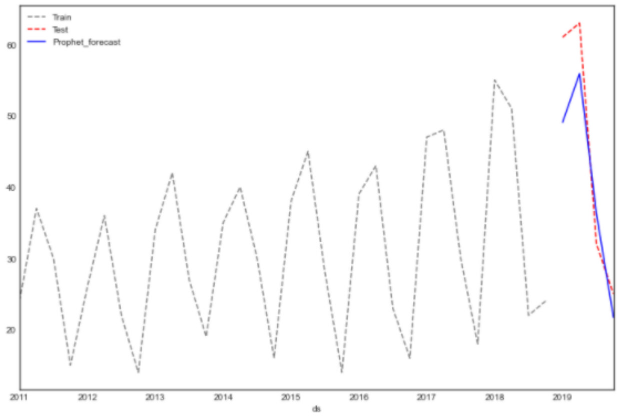

Figure 5.14: Forecasts from Prophet model

to the automated process of generating forecasts. The performance metrics was calculated and we obtained the RMSE value as **7.4** and R^2 value as **0.81**.

5.7.6 Multilayer Perceptron (MLP)

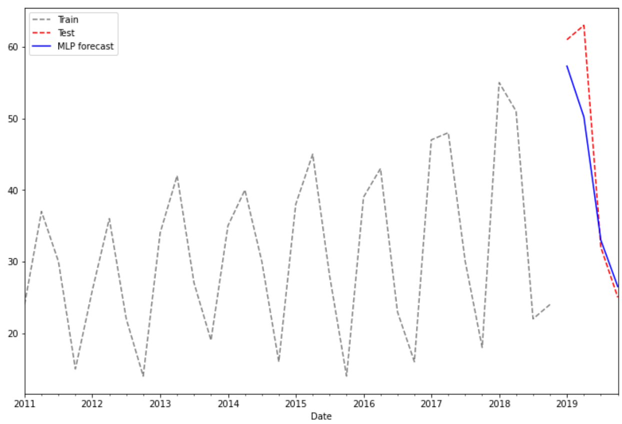

Figure 5.15: Forecasts from MLP model

The results of the prediction using the MLP model is shown in the figure 5.15. As we can see from the plot, the second quarter has been under-predicted compared to the other three quarters in 2019. It is difficult to explain the presence of certain variables in the final model due to the 'black-box' nature of the model. Consequently, the fitted values were also not available. The performance metrics for the model were calculated and we obtained the RMSE value as **6.7** and R^2

value as **0.84**.

5.7.7 Long Short Term Memory (LSTM)

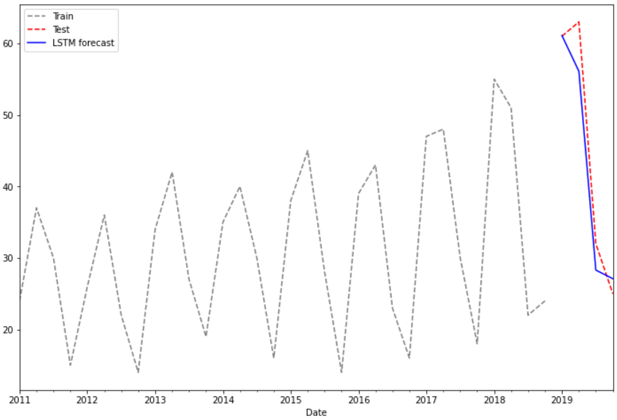

Figure 5.16: Forecasts from LSTM model

The results of the prediction using the LSTM model is shown in the figure 5.16. As we can see from the plot, the second and third quarters have been under-predicted compared to the other two quarters in 2019. It is difficult to explain the presence of certain variables in the final model due to the 'black-box' nature of the model. Consequently, the fitted values were also not available. The performance metrics for the model were calculated and we obtained the RMSE value as **4.1** and R^2 value as **0.94**.

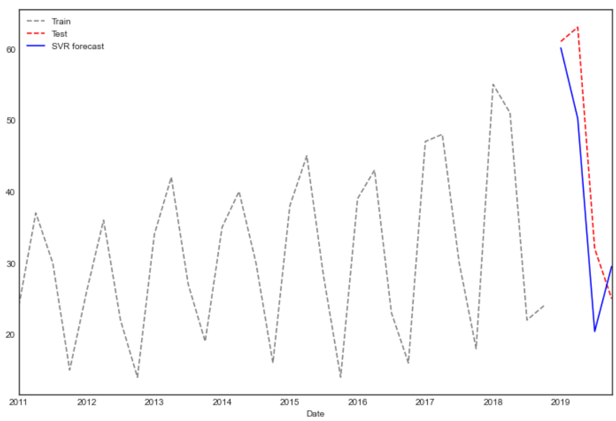

Figure 5.17: Forecasts from SVR model

5.7.8 Support Vector Regressor (SVR)

The results of the prediction are shown in the figure 5.17. From the graph, we noticed that the first and last quarter was predicted close to the actual observed value while the other two quarters had bigger differences between the predicted values and the actual observed values. It is difficult to explain the presence of certain variables in the final model due to the 'black-box' nature of the model. Consequently, the fitted values were also not available. The performance metrics were calculated and we obtained the RMSE value as **8.9** and R^2 value as **0.72**.

Chapter 6

Discussions

This chapter discusses the findings from the literature study and quantitative modeling to answer the research questions.

The case company, Scale AQ, had a long replenishment lead-time for one of their crucial component, that is brackets. The brackets are considered as a functional product and the demand of a functional product is generally believed to be stable and predictable (Chopra and Meindl 2016). But the demand of the brackets was unpredictable because it depends on the demand of the final product, cage-P. Sanderson and Cox (2008) also argued that functional products in a low-volume complex environment exhibits demand unpredictability. To mitigate the supply-demand mismatch, we have chosen to improve the demand forecasting capability of the case company.

The objective of this study was to find a suitable forecasting method that could handle limited data and generate forecasts for a short-medium forecast horizon. To find a suitable forecasting method, various time-series forecasting methods were investigated. Two research questions were formulated to support us in achieving the research objective. We begin with attempting to answer the first research question which is ***"What are the different state-of-the-art traditional and advanced forecasting methods that can be employed on short univariate time-series data?"***

In order to answer the first research question, we utilized systematic literature review to find the forecasting models. According to Feizabadi (2020), plenty of attention was contributed towards artificial intelligence to improve the demand forecastability of an organisation in the recent years. Traditional demand forecasting methods accommodates only few factors such as trend, seasonality, cyclical behaviour and so on, that affect demand. On the contrary, advanced methods can combine artificial intelligence and machine learning algorithms to analyze the demand data and can account for the extensive number of casual factors along with non-linear relationship between the predictor and target variables. Though advanced methods are generally labelled data-hungry methods, we noticed that some of them performed well with limited demand data in few literature articles (refer table 4.3). Due to the inconclusive results in selecting a demand forecasting method for limited data (or short time-series), we have investigated the performances of both traditional and advanced methods in handling limited demand data.

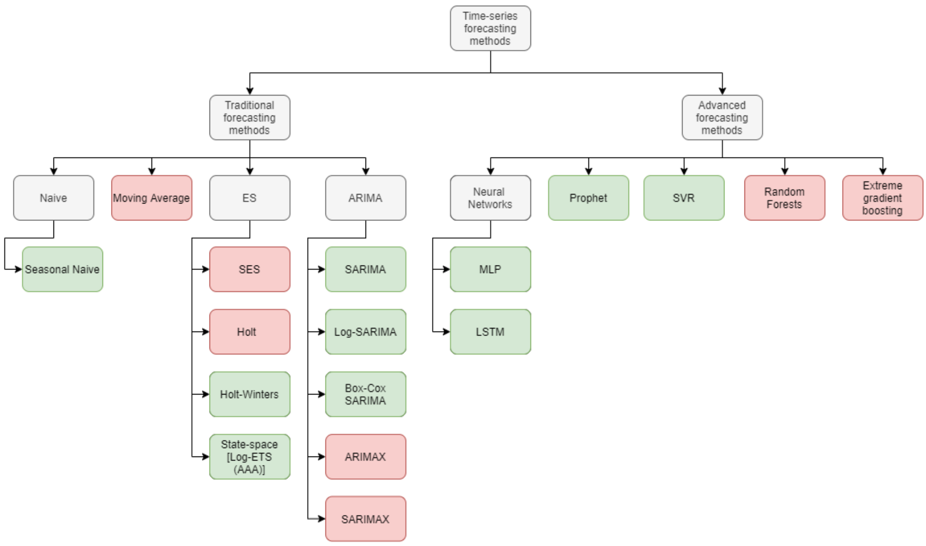

Figure 6.1: Selected time-series forecasting models

Most of the literature that we came across was about energy demand forecasting with long time-series (consisted of several hundreds or thousands of data-points). We found mainly four traditional methods and five advanced methods along with their variants (refer figure 6.1) that were the most popularly studied methods for demand forecasting. The characteristics, advantages and disadvantages of each model were also discussed to further support the filtering process. From the mentioned models, only some of them were chosen for quantitative modeling based on the selection criteria mentioned in the section 4.4. Four traditional models (and its variants) and four advanced models were selected for the analysis. Some of the model variants in the traditional methods were a result of data transformations performed on the original series. For instance, in Log-SARIMA, log transformation was performed on the original series. The models colored in green in the figure 6.1 were the ones selected for modeling. The other models indicated in red color in the figure 6.1 were not included for modeling because they did not qualify the selection requirements. Moving average, SES and Holt models were not considered because they could not handle seasonality behaviour. ARIMAX and SARIMAX models were not considered because we do not use exogenous information in our study and it is beyond the scope of the study. Random forests and XGBoost were not selected for the quantitative analysis due to the fact that they required a minimum of 50 data-points to perform well (Pedregosa, G. Varoquaux, et al. 2011). The list of models chosen for quantitative modeling are: seasonal naïve, Holt-Winters, state-space (ETS), SARIMA (and its data transformed variants), MLP, LSTM, Prophet and SVR. The previously stated models answers our first research question.

Now we attempt to answer the second research question which is ***”Do advanced time-series forecasting methods perform better than traditional time-series forecasting methods when dealing with short univariate time-series data?”***

In order to answer the research question, quantitative analysis was performed in Python environment with the chosen forecasting models using the demand data from the case-company. Each model was subjected to the same data set consisting of 32 quarterly sales observations from 2011-2018 and were evaluated on a test set (consisting of 4 quarters) using RMSE and R^2 as the performance metrics. Data transformation was applied to most of the models before modeling and all models were optimised using grid-search and hand-tuning techniques. For traditional methods, the values of the variables of each model were also reported along with residual plots. On the contrary, neither the values of the variables of each model in the advanced methods nor the residuals plots were not reported due to their *black-box* nature.

The performance metrics of all the models is shown in the table 6.1. The performance metrics were also plotted as shown in the figures 6.2 & 6.3 to support in comparison against each other. The performance of a model is considered to be superior if it has a lower RMSE value and a R^2 value closer to one. Both the performance metrics revealed that Log-SARIMA was the best model among the traditional methods and LSTM was the best model among the advanced methods. The bold values in the table 6.1 indicate the previously discussed result. From the plots and the table 6.1, it was evident that LSTM was the overall best forecasting model which was very closely followed by Log-SARIMA, and the

	Model	RMSE	R^2
Traditional methods	Seasonal naive	8.4	0.76
	Holt-Winters	5.3	0.9
	Log-ETS (AAA)	5.5	0.89
	SARIMA	6.0	0.87
	Log-SARIMA	**4.4**	**0.93**
	Box-Cox SARIMA	5.8	0.88
Advanced methods	Prophet	7.4	0.81
	MLP	6.7	0.84
	LSTM	**4.1**	**0.94**
	SVR	8.9	0.72

Table 6.1: Performance metrics of all the selected models

overall worst forecasting model was SVR. LSTM performed slightly better than SARIMA models due to the availability of large number of hyperparameters and its ability to handle non-linear trends.

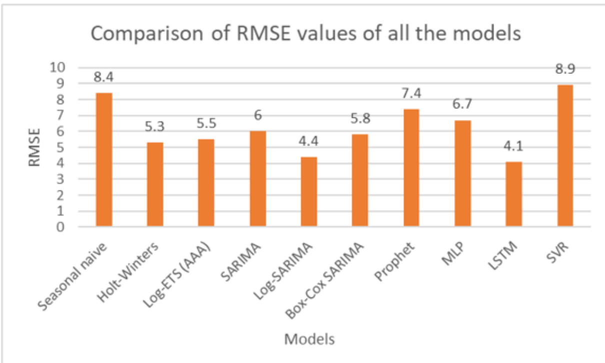

Figure 6.2: RMSE values of all the models

Seasonal naïve model was also used as a base model to benchmark the performances of the other complex models. All the models, except SVR, performed better than the base model. Since the currently employed time-series forecasting method (seasonal naïve) did not provide accurate forecasts for the company, we suggest that seasonal naïve could be replaced with any of the chosen state-of-the-art models (except SVR) to improve the forecasting performance. We noticed a minimum improvement of 12% in forecasting performance when Prophet model was employed and maximum of 51% when LSTM model was employed. The other models except SVR contributed an improvement in forecasting performance within the previously mentioned range. The second best performing model, SARIMA and its variants, resulted in an improvement of 29-48%

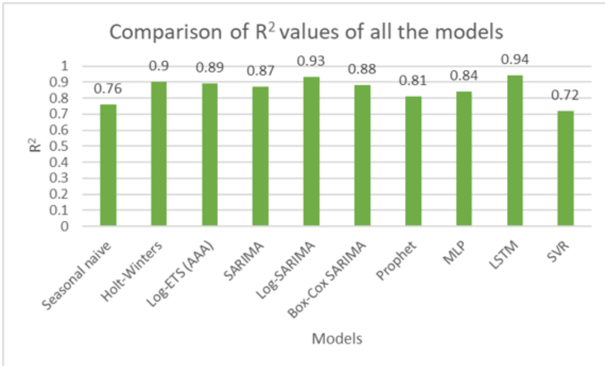

Figure 6.3: R^2 values of all the models

compared to seasonal naïve's forecasting performance. In general, we noticed that most of the traditional methods outperformed the advanced methods, except LSTM. Hence, traditional methods are more reliable for generating forecasts with short univariate time-series. This result leads us to answering the second research question.

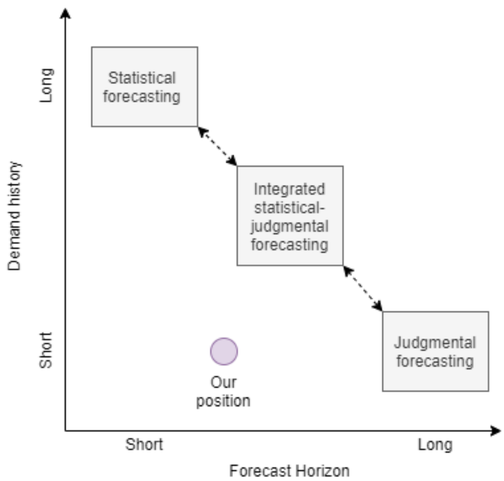

Figure 6.4: Type of forecasting in relation to demand history and forecast horizon (adapted from Syntetos et al. (2016))

Domain knowledge, such as new product introductions, R&D milestones and so on due to which the demand is provoked, cannot be captured by a forecasting model or exogenous variables. Also, the business experts have more information in the near future regarding the current pipeline of orders that will be arriving.

The model derived forecasts could support the sales and operations team in an organisation in evaluating the soft orders received from their customers and also in providing information on longer term sales trends. Domain knowledge of business experts is vital especially in a low-volume project environment and it is useful to adjust the derived model forecasts based on the situation. The previous discussion is depicted in the form of a graph in the figure 6.4. The purple dot in the figure 6.4 indicates our problem position. We have short demand history and have to generate forecasts for short-medium term horizon (one year). Since our problem is positioned quite close to integrated statistical-judgmental forecasts in the graph, we suggest that complementing model derived (statistical) forecasts with managerial judgments could improve the forecasting performance (Syntetos et al. 2016). Feizabadi (2020) and Salais-Fierro et al. (2020) also demonstrated an increase in efficiency in the planning process when managers make decisions based on more than one method.

Chapter 7

Conclusion

This chapter presents the conclusion of the research which is followed by the limitations faced in the study and finally suggestions for future work.

The objective of this comparative study was to investigate various time-series forecasting methods that could handle limited data, the necessary data transformation techniques and finally, performance metrics to evaluate the results of the modeling. The model derived forecasts are intended to guide the managers in the demand and supply planning process of Scale AQ. In order to achieve this objective, two research questions were framed. They were answered using exploratory case study and systematic literature review as the research methods.

There are a plethora of time-series forecasting models that were developed for demand forecasting. Among those models, only a few models could handle limited data (short time-series). These models were selected based on some requirements and were studied in depth. A total of 8 models were selected for the analysis, namely,seasonal naïve, Holt-Winters, state-space (ETS), SARIMA (and its data transformed variants), MLP, LSTM, Prophet and SVR. The first four models belonged to traditional methods and the remaining four models belonged to advanced methods. The data utilized for modeling was historical demand data of cage-P. The data was scrutinized for irregularities and the necessary data transformations were performed before modeling. The selected models utilized the same data to generate forecasts for 4 quarters. Several model configurations were tested and the suitable parameters for each model were chosen from grid-searching and hand-tuning. They were evaluated using the performance metrics, RMSE and R^2.

The models were compared against each other. It was found that LSTM was the overall best performing model and SVR was the overall worst performing model. SARIMA model with log transformed data was the best performing model in traditional methods with an improvement of 48% compared to seasonal naïve forecasts (which is the currently employed model at the case company). LSTM model resulted in a maximum improvement of 51% and Prophet model resulted in a minimum improvement of 12% compared to the seasonal naïve model. The other models except SVR contributed towards improvement in forecasting performance within the previously mentioned range. In general, we also noticed that the traditional methods performed better than most of the advanced methods. Thus, we concluded that traditional methods were superior compared to advanced methods when dealing with short univariate time-series.

7.1 Contribution

This research contributes to both academia and case company. For academia, this research contributes to the field by comparison of various traditional and advanced forecasting methods that are optimized to handle short time-series. Though there has been comprehensive research on comparison of traditional and advanced forecasting models, most of the research utilised longer time series (Bouktif et al. 2018; Carbonneau et al. 2008; Divina et al. 2019; Parmezan et al. 2019; Yenidoğan et al. 2018; X. Zhang et al. 2014). Also, this study included some of the latest state-of-the-art forecasting models such as Prophet and LSTM. To the best of the knowledge of the author, we did not come across any article that neither compared all the mentioned state-of-the-art forecasting models nor demonstrated a detailed model description for all the models. This study suggests a suitable method for forecasting short univariate time series to a practitioner.

For Scale AQ, the model forecasts serve as a tool to the sales and operations team to evaluate the soft orders (or handshake plan). They also help the company with replacing the currently employed statistical forecasting with a state-of-the-art forecasting method and consequently ensuring better forecasting performance. Due to the superior performance of traditional models in forecasting short time-series, we recommend that SARIMA model could be used by Scale AQ for forecasting the demand of cage-P and other products. Compared to seasonal naïve, an improvement of 29-48% in forecasting performance was obtained by using the SARIMA model and its variants. The model configuration code has to be triggered to select the appropriate configuration that is applicable for each product. The model configuration code has to be run every year to prevent it from being outdated because the demand pattern changes over time. Since each product has its own unique demand data associated with it and the model configuration is dependent on the time-series data; therefore, the model configuration code is run separately for each of them. Once the actual demand is realized, they should be updated in the training data and consequently, the model generates forecasts for the required forecast horizon. We also recommend to augment managerial judgments with the SARIMA model forecasts due to the importance of domain knowledge in a low-volume complex environment, which might consequently lead to further improvement in forecast accuracy.

7.2 Limitations

There were two main limitations involved in this research. Firstly, the author is not well-versed in the field of data-science and therefore he used only grid search technique for tuning the hyperparameters. Further tuning of the hyperparameters using other techniques or other popular hyperparameter tuning libraries in Python might lead to more accurate forecasts.

Secondly, building a robust forecasting model with limited historical sales data. There were only 36 data points with only 32 data points used as a train set. This made the training of a robust forecasting model extremely challenging especially with the advanced models. Although some of the advanced models provided superior performance in the literature, generally most of them tend to overfit on the training data and perform poorly on test data. For instance, SVR performed well in many literature articles with limited data but performed poorly in our analysis. Several hundreds or even thousands of data points might be required to train a robust advanced forecasting model. With the increase in the availability of data, the performance of the advanced models would also improve drastically.

7.3 Future Work

Prior research (Makridakis et al. 2020; Xu et al. 2019; G Peter Zhang 2003) suggests the idea of combining models to create hybrid models that could improve the forecasting performance. A hybrid model could be a combination of two advanced models or a traditional model and an advanced model. As a future work, hybrid models in demand forecasting using short univariate time-series can be investigated.

Multivariate analysis of the time-series forecasting methods using short time-series could be investigated. Multivariate analysis allows the forecaster to add more predictor variables that could possibly influence the demand of cages.

Bibliography

Abbasimehr, Hossein, Mostafa Shabani, and Mohsen Yousefi (May 2020). "An optimized model using LSTM network for demand forecasting". en. In: *Computers & Industrial Engineering* 143, p. 106435. ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106435. URL: http://www.sciencedirect.com/science/article/pii/S0360835220301698.

Abdel-Aal, R. E. (May 2008). "Univariate modeling and forecasting of monthly energy demand time series using abductive and neural networks". en. In: *Computers & Industrial Engineering* 54.4, pp. 903–917. ISSN: 0360-8352. DOI: 10.1016/j.cie.2007.10.020. URL: http://www.sciencedirect.com/science/article/pii/S0360835207002471.

Ahmed, Nesreen K et al. (2010). "An empirical comparison of machine learning models for time series forecasting". In: *Econometric Reviews* 29.5-6, pp. 594–621.

Akpinar, Mustafa and Nejat Yumusak (Sept. 2016). "Year Ahead Demand Forecast of City Natural Gas Using Seasonal Time Series Methods". en. In: *Energies* 9.9. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 727. DOI: 10.3390/en9090727. URL: https://www.mdpi.com/1996-1073/9/9/727.

Awad, Mariette and Rahul Khanna (2015). "Support Vector Regression". en. In: *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*. Ed. by Mariette Awad and Rahul Khanna. Berkeley, CA: Apress, pp. 67–80. ISBN: 978-1-4302-5990-9. DOI: 10.1007/978-1-4302-5990-9_4. URL: https://doi.org/10.1007/978-1-4302-5990-9_4.

Bakshi, Chaya (June 2020). *Random Forest Regression*. en. URL: https://levelup.gitconnected.com/random-forest-regression-209c0f354c84.

Bandara, Kasun et al. (2019). "Sales demand forecast in e-commerce using a long short-term memory neural network methodology". In: *International Conference on Neural Information Processing*. Springer, pp. 462–474.

Bouktif, Salah et al. (July 2018). "Optimal Deep Learning LSTM Model for Electric Load Forecasting using Feature Selection and Genetic Algorithm: Comparison with Machine Learning Approaches †". en. In: *Energies* 11.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1636. DOI: 10.3390/en11071636. URL: https://www.mdpi.com/1996-1073/11/7/1636.

Box, George EP, Gwilym M Jenkins, and Gregory C Reinsel (2011). *Time series analysis: forecasting and control*. Vol. 734. John Wiley & Sons.

Breiman, Leo (Oct. 2001). "Random Forests". en. In: *Machine Learning* 45.1, pp. 5–32. ISSN: 1573-0565. DOI: 10.1023/A:1010933404324.

Brownlee, Jason (2017). *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery.

– (2018). *Deep learning for time series forecasting: Predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery.

Buer, Sven-Vegard, Jan Ola Strandhagen, and Felix TS Chan (2018). "The link between Industry 4.0 and lean manufacturing: mapping current research and establishing a research agenda". In: *International Journal of Production Research* 56.8, pp. 2924–2940.

Buxton, E. et al. (Dec. 2019). "An Auto Regressive Deep Learning Model for Sales Tax Forecasting from Multiple Short Time Series". In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1359–1364. DOI: 10.1109/ICMLA.2019.00221.

Cankurt, Selcuk and Abdülhamit Subaşi (2016). "Tourism demand modelling and forecasting using data mining techniques in multivariate time series: a case study in Turkey". In: *Turkish Journal of Electrical Engineering & Computer Sciences* 24.5, pp. 3388–3404.

Carbonneau, Real, Kevin Laframboise, and Rustam Vahidov (Feb. 2008). "Application of machine learning techniques for supply chain demand forecasting". en. In: *European Journal of Operational Research* 184.3, pp. 1140–1154. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2006.12.004. URL: http://www.sciencedirect.com/science/article/pii/S0377221706012057.

Carpenter, Bob et al. (Jan. 2017). "Stan: A Probabilistic Programming Language". en. In: *Journal of Statistical Software* 76.1. Number: 1, pp. 1–32. ISSN: 1548-7660. DOI: 10.18637/jss.v076.i01. URL: https://www.jstatsoft.org/index.php/jss/article/view/v076i01.

Chanklan, Ratiporn et al. (2018). "Runoff Prediction with a Combined Artificial Neural Network and Support Vector Regression". In: *International Journal of Machine Learning and Computing* 8.1.

Chen, Tianqi and Carlos Guestrin (Aug. 2016). "XGBoost: A Scalable Tree Boosting System". en. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, pp. 785–794. ISBN: 978-1-4503-4232-2. DOI: 10.1145/2939672.2939785. URL: https://dl.acm.org/doi/10.1145/2939672.2939785.

Chen, Tianqi, T He, and M Benesty (2018). *XGBoost Documentation*.

Chollet, François et al. (2018). "Keras: The python deep learning library". In: *Astrophysics Source Code Library*, ascl–1806.

Chopra, Sunil and Peter Meindl (2016). *Supply chain management: strategy, planning, and operation*. eng. 6th ed., Global ed. Pages: 528. Harlow: Pearson. ISBN: 978-1-292-09356-7.

Claveria, Oscar and Salvador Torra (Jan. 2014). "Forecasting tourism demand to Catalonia: Neural networks vs. time series models". en. In: *Economic Modelling* 36, pp. 220–228. ISSN: 0264-9993. DOI: 10.1016/j.econmod.2013.09.024. URL: http://www.sciencedirect.com/science/article/pii/S0264999313003842.

Creswell, John W and J David Creswell (2017). *Research design: Qualitative, quantitative, and mixed methods approaches*. Sage publications.

Croom, Simon (2010). "Introduction to research methodology in operations management". In: *Researching operations management*. Routledge, pp. 56–97.

De Gooijer, Jan G. and Rob J. Hyndman (Jan. 2006). "25 years of time series forecasting". en. In: *International Journal of Forecasting*. Twenty five years of forecasting 22.3, pp. 443–473. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2006.01.001. URL: http://www.sciencedirect.com/science/article/pii/S0169207006000021.

Deb, Chirag et al. (July 2017). "A review on time series forecasting techniques for building energy consumption". en. In: *Renewable and Sustainable Energy Reviews* 74, pp. 902–924. ISSN: 1364-0321. DOI: 10.1016/j.rser.2017.02.085. URL: http://www.sciencedirect.com/science/article/pii/S1364032117303155.

"Concurrent, Performance-Based Methodology for Increasing the Accuracy and Certainty of Short-Term Neural Prediction Systems" (2019). In: 2019. Ed. by Vlado Delic. Num Pages: 12 Place: New York, United Kingdom Publisher: Hindawi Limited. ISSN: 16875265. DOI: http://dx.doi.org/10.

1155/2019/9323482. URL: https://www.proquest.com/docview/2209522263/abstract/F63D709A28EF469CPQ/1.

Demolli, Halil et al. (Oct. 2019). "Wind power forecasting based on daily wind speed data using machine learning algorithms". en. In: *Energy Conversion and Management* 198, p. 111823. ISSN: 0196-8904. DOI: 10.1016/j.enconman.2019.111823. URL: http://www.sciencedirect.com/science/article/pii/S0196890419308052.

Divina, Federico et al. (Jan. 2019). "A Comparative Study of Time Series Forecasting Methods for Short Term Electric Energy Consumption Prediction in Smart Buildings". en. In: *Energies* 12.10. Number: 10 Publisher: Multidisciplinary Digital Publishing Institute, p. 1934. DOI: 10.3390/en12101934. URL: https://www.mdpi.com/1996-1073/12/10/1934.

Dudek, Grzegorz (Jan. 2015). "Short-Term Load Forecasting Using Random Forests". In: *Advances in Intelligent Systems and Computing*. Vol. 323. Journal Abbreviation: Advances in Intelligent Systems and Computing, pp. 821–828. ISBN: 978-3-319-11309-8. DOI: 10.1007/978-3-319-11310-4_71.

Feizabadi, Javad (Aug. 2020). "Machine learning demand forecasting and supply chain performance". In: *International Journal of Logistics Research and Applications* 0.0. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/13675567.2020.18 pp. 1–24. ISSN: 1367-5567. DOI: 10.1080/13675567.2020.1803246. URL: https://doi.org/10.1080/13675567.2020.1803246.

G Smith, Taylor (Dec. 2020). *pmdarima: Python's forecast::auto.arima equivalent*. URL: http://alkaline-ml.com/pmdarima.

Gonzalez-Romera, E., M. A. Jaramillo-Moran, and D. Carmona-Fernandez (2006). "Monthly Electric Energy Demand Forecasting Based on Trend Extraction". In: *IEEE Transactions on Power Systems* 21.4, pp. 1946–1953. DOI: 10.1109/TPWRS.2006.883666.

Greff, K. et al. (2017). "LSTM: A Search Space Odyssey". In: *IEEE Transactions on Neural Networks and Learning Systems* 28.10, pp. 2222–2232. DOI: 10.1109/TNNLS.2016.2582924.

Harris, Charles R. et al. (Sept. 2020). "Array programming with NumPy". In: *Nature* 585.7825, pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

Haykin, Simon S et al. (2009). *Neural networks and learning machines/Simon Haykin.*

Hunter, J. D. (2007). "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3, pp. 90–95. DOI: 10.1109/MCSE.2007.55.

Hyndman, Rob J and George Athanasopoulos (2018). *Forecasting: Principles and Practice*. 2nd edition. URL: https://Otexts.com/fpp2/.

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

Hyndman, Rob J. and Anne B. Koehler (Oct. 2006). "Another look at measures of forecast accuracy". en. In: *International Journal of Forecasting* 22.4, pp. 679–688. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2006.03.001. URL: http://www.sciencedirect.com/science/article/pii/S0169207006000239.

Ilbeigi, M., B. Ashuri, and A. Joukar (Jan. 2017). "Time-Series Analysis for Forecasting Asphalt-Cement Price". EN. In: *Journal of Management in Engineering* 33.1. Publisher: American Society of Civil Engineers, p. 04016030. ISSN: 1943-5479. DOI: 10.1061/(ASCE)ME.1943-5479.0000477. URL: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29ME.1943-5479.0000477.

Ismail Fawaz, Hassan et al. (July 2019). "Deep learning for time series classification: a review". In: *Data Mining and Knowledge Discovery* 33.4. Publisher: Springer, pp. 917–963. DOI: 10.1007/s10618-019-00619-1. URL: https://hal.archives-ouvertes.fr/hal-02365025.

Jere, Stanley, Bornwell Kasense, and Obvious Chilyabanyama (Feb. 2017). "Forecasting Foreign Direct Investment to Zambia: A Time Series Analysis". en. In: *Open Journal of Statistics* 07.01. Number: 01 Publisher: Scientific Research Publishing, p. 122. DOI: 10.4236/ojs.2017.71010. URL: http://www.scirp.org/journal/PaperInformation.aspx?PaperID=74499&#abstract.

Johannesen, Nils Jakob, Mohan Kolhe, and Morten Goodwin (May 2019). "Relative evaluation of regression tools for urban area electrical energy demand forecasting". en. In: *Journal of Cleaner Production* 218, pp. 555–564. ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2019.01.108. URL: http://www.sciencedirect.com/science/article/pii/S0959652619301192.

Jung-Pin, Lai et al. (2020). "A Survey of Machine Learning Models in Renewable Energy Predictions". English. In: *Applied Sciences* 10.17. Num Pages: 5975 Place: Basel, Switzerland Publisher: MDPI AG, p. 5975. DOI: http://dx.doi.org/10.3390/app10175975. URL: https://www.proquest.com/docview/2439491946/abstract/FDA76DFAE9BC4F43PQ/62.

Kalekar, Prajakta S. (2004). "Time series Forecasting using Holt-Winters Exponential Smoothing". en. In: URL: /paper/Time-series-Forecasting-using-Holt-Winters-Kalekar/1eae285bd8d62eaf9dc37e8d2351845ba

Karlsson, Christer (May 2016). *Research Methods for Operations Management*. en. 2nd ed. Second Edition. — New York : Routledge, 2016. — Revised: Routledge. ISBN: 978-1-315-67142-0. DOI: 10.4324/9781315671420. URL: https://www.taylorfrancis.com/books/9781315671420.

Kim, Seul-Gi, Jae-Yoon Jung, and Min Kyu Sim (2019). "A two-step approach to solar power generation prediction based on weather data using machine learning". In: *Sustainability* 11.5, p. 1501.

Kothari, C. R. (2004). *Research Methodology: Methods and Techniques*. en. New Age International. ISBN: 978-81-224-1522-3.

Kusiak, Andrew, Haiyang Zheng, and Zhe Song (2009). "Short-term prediction of wind farm power: A data mining approach". In: *IEEE Transactions on energy conversion* 24.1, pp. 125–136.

Lahouar, A. and J. Ben Hadj Slama (Aug. 2017). "Hour-ahead wind power forecast based on random forests". en. In: *Renewable Energy* 109, pp. 529–541. ISSN: 0960-1481. DOI: 10.1016/j.renene.2017.03.064. URL: http://www.sciencedirect.com/science/article/pii/S0960148117302550.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos (Mar. 2018). "Statistical and Machine Learning forecasting methods: Concerns and ways forward". en. In: *PLOS ONE* 13.3. Publisher: Public Library of Science, e0194889. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0194889.

– (Jan. 2020). "The M4 Competition: 100,000 time series and 61 forecasting methods". en. In: *International Journal of Forecasting*. M4 Competition 36.1, pp. 54–74. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2019.04.014.

Martínez-Álvarez, Francisco et al. (Nov. 2015). "A Survey on Data Mining Techniques Applied to Electricity-Related Time Series Forecasting". en. In: *Energies* 8.11. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, pp. 13162–13193. DOI: 10.3390/en81112361. URL: https://www.mdpi.com/1996-1073/8/11/12361.

McKinney, Wes et al. (2010). "Data structures for statistical computing in python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX, pp. 51–56.

Meredith, Jack (1998). "Building operations management theory through case and field research". In: *Journal of operations management* 16.4, pp. 441–454.

Nahmias, Steven and Tava Lennon Olsen (Oct. 2020). *Production and Operations Analytics: Eighth Edition*. en. Google-Books-ID: ctQBEAAAQBAJ. Waveland Press. ISBN: 978-1-4786-4630-3.

Naim, Iram and Tripti Mahara (May 2018). "Comparative Analysis of Univariate Forecasting Techniques for Industrial Natural Gas Consumption". In: *International Journal of Image, Graphics and Signal Processing* 10, pp. 33–44. DOI: 10.5815/ijigsp.2018.05.04.

Naim, Iram, Tripti Mahara, and Sharfuddin Ahmed Khan (Jan. 2020). "Ranking of univariate forecasting techniques for seasonal time series using analytical hierarchy process". In: *International Journal of Industrial and Systems Engineering* 35.2. Publisher: Inderscience Publishers, pp. 196–215. ISSN: 1748-5037. DOI: 10.1504/IJISE.2020.107581. URL: https://www.inderscienceonline.com/doi/abs/10.1504/IJISE.2020.107581.

Nau, Robert (2020). *Statistical forecasting: notes on regression and time series analysis*. URL: https://people.duke.edu/~rnau/411home.htm.

Olah, Christopher (Aug. 2015). *Understanding LSTM Networks – colah's blog*. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on Dec. 27, 2020).

Padhan, Purna (Jan. 2012). "Use of univariate time series models for forecasting cement productions in India". In: *International Research Journal of Finance and Economics* 83, pp. 167–179.

Papacharalampous, Georgia, Hristos Tyralis, and Demetris Koutsoyiannis (Dec. 2018). "Univariate Time Series Forecasting of Temperature and Precipitation with a Focus on Machine Learning Algorithms: a Multiple-Case Study from Greece". en. In: *Water Resources Management* 32.15, pp. 5207–5239. ISSN: 1573-1650. DOI: 10.1007/s11269-018-2155-6. URL: https://doi.org/10.1007/s11269-018-2155-6.

Papastefanopoulos, Vasilis, Pantelis Linardatos, and Sotiris Kotsiantis (Jan. 2020). "COVID-19: A Comparison of Time Series Methods to Forecast Percentage of Active Cases per Population". en. In: *Applied Sciences* 10.11. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, p. 3880. DOI: 10.3390/app10113880. URL: https://www.mdpi.com/2076-3417/10/11/3880.

Parmezan, Antonio Rafael Sabino, Vinicius M. A. Souza, and Gustavo E. A. P. A. Batista (May 2019). "Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model". en. In: *Information Sciences* 484, pp. 302–337. ISSN: 0020-0255. DOI: 10.1016/j.ins.2019.01.076. URL: http://www.sciencedirect.com/science/article/pii/S0020025519300945.

Pavlyshenko, Bohdan M. (Mar. 2019). “Machine-Learning Models for Sales Time Series Forecasting”. en. In: *Data* 4.1. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute, p. 15. DOI: 10.3390/data4010015. URL: https://www.mdpi.com/2306-5729/4/1/15.

Pedregosa, Fabian, G. Varoquaux, et al. (2011). *Choosing the right estimator — scikit-learn 0.23.2 documentation*. URL: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html.

Pedregosa, Fabian, Gaël Varoquaux, et al. (2011). “Scikit-learn: Machine learning in Python”. In: *the Journal of machine Learning research* 12, pp. 2825–2830.

Peng, Wen-Yi and Ching-Wu Chu (Oct. 2009). “A comparison of univariate methods for forecasting container throughput volumes”. en. In: *Mathematical and Computer Modelling* 50.7, pp. 1045–1057. ISSN: 0895-7177. DOI: 10.1016/j.mcm.2009.05.027. URL: http://www.sciencedirect.com/science/article/pii/S0895717709001794.

Phi, Michael (June 2020). *Illustrated Guide to LSTM’s and GRU’s: A step by step explanation*. en. URL: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Ramos, Patrícia and Jose Manuel Oliveira (2016). “A Procedure for Identification of Appropriate State Space and ARIMA Models Based on Time-Series Cross-Validation”. English. In: *Algorithms* 9.4. Num Pages: 76 Place: Basel, Switzerland Publisher: MDPI AG, p. 76. DOI: http://dx.doi.org/10.3390/a9040076. URL: https://www.proquest.com/docview/1858321991/abstract/99A14A425F834CFFPQ/2.

Remus, William and Marcus O’Connor (2001). “Neural Networks for Time-Series Forecasting”. en. In: *Principles of Forecasting: A Handbook for Researchers and Practitioners*. Ed. by J. Scott Armstrong. International Series in Operations Research & Management Science. Boston, MA: Springer US, pp. 245–256. ISBN: 978-0-306-47630-3. DOI: 10.1007/978-0-306-47630-3_12. URL: https://doi.org/10.1007/978-0-306-47630-3_12.

Rivero, Cristian Rodrìguez et al. (Jan. 2017). “Energy Associated Tuning Method for Short-Term Series Forecasting by Complete and Incomplete Datasets”. en. In: *Journal of Artificial Intelligence and Soft Computing Research* 7.1. Publisher: Sciendo Section: Journal of Artificial Intelligence and Soft Computing Research, pp. 5–16. DOI: 10.1515/jaiscr-2017-0001. URL: https://content.sciendo.com/view/journals/jaiscr/7/1/article-p5.xml.

Salais-Fierro, Tomas Eloy et al. (Jan. 2020). "Demand Prediction Using a Soft-Computing Approach: A Case Study of Automotive Industry". en. In: *Applied Sciences* 10.3. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 829. DOI: 10.3390/app10030829. URL: https://www.mdpi.com/2076-3417/10/3/829.

Samal, K Krishna Rani et al. (2019). "Time Series based Air Pollution Forecasting using SARIMA and Prophet Model". In: *Proceedings of the 2019 International Conference on Information Technology and Computer Communications*, pp. 80–85.

Samsudin, R, A Shabri, and P Saad (2010). "A comparison of time series forecasting using support vector machine and artificial neural network model". In: *JApSc* 10.11, pp. 950–958.

Sanderson, Joe and Andrew Cox (2008). "The challenges of supply strategy selection in a project environment: evidence from UK naval shipbuilding". In: *Supply Chain Management: An International Journal*.

Santamaría-Bonfil, G., A. Reyes-Ballesteros, and C. Gershenson (Jan. 2016). "Wind speed forecasting for wind farms: A method based on support vector regression". en. In: *Renewable Energy* 85, pp. 790–809. ISSN: 0960-1481. DOI: 10.1016/j.renene.2015.07.004. URL: http://www.sciencedirect.com/science/article/pii/S0960148115301014.

Seabold, Skipper and Josef Perktold (2010). "Statsmodels: Econometric and statistical modeling with python". In: *Proceedings of the 9th Python in Science Conference*. Vol. 57. Austin, TX, p. 61.

Sharifzadeh, M. et al. (2019). "Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian Process Regression". In: DOI: 10.1016/J.RSER.2019.03.040.

Shih, Han and Suchithra Rajendran (Sept. 2019). "Comparison of Time Series Methods and Machine Learning Algorithms for Forecasting Taiwan Blood Services Foundation's Blood Supply". en. In: *Journal of Healthcare Engineering*. ISSN: 2040-2295 Pages: e6123745 Publisher: Hindawi Volume: 2019. DOI: https://doi.org/10.1155/2019/6123745. URL: https://www.hindawi.com/journals/jhe/2019/6123745/.

Stoica, P. and Y. Selen (2004). "Model-order selection: a review of information criterion rules". In: *IEEE Signal Processing Magazine* 21.4, pp. 36–47. DOI: 10.1109/MSP.2004.1311138.

Swalin, Alvira (July 2018). *Choosing the Right Metric for Evaluating Machine Learning Models — Part 1*. en. URL: https://medium.com/usfmsds/choosing-the-right-metric-for-machine-learning-models-part-1-a99d7d7414e4 (visited on Jan. 4, 2021).

Syntetos, Aris A. et al. (July 2016). "Supply chain forecasting: Theory, practice, their gap and the future". en. In: *European Journal of Operational Research* 252.1, pp. 1–26. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2015.11.010. URL: http://www.sciencedirect.com/science/article/pii/S0377221715010231.

Tanizaki, Takashi et al. (2019). "Demand forecasting in restaurants using machine learning and statistical analysis". en. In: *Procedia CIRP* 79, pp. 679–683. ISSN: 22128271. DOI: 10.1016/j.procir.2019.02.042. URL: https://linkinghub.elsevier.com/retrieve/pii/S2212827119301568.

Taylor, James W. (2008). "A Comparison of Univariate Time Series Methods for Forecasting Intraday Arrivals at a Call Center". In: *Management Science* 54.2. Publisher: INFORMS, pp. 253–265. ISSN: 0025-1909. URL: https://www.jstor.org/stable/20122381.

Taylor, Sean and Ben Letham (2018). *Automatic Forecasting Procedure*. URL: http://facebook.github.io/prophet/.

Torres-Barrán, Alberto, Álvaro Alonso, and José R. Dorronsoro (Jan. 2019). "Regression tree ensembles for wind energy and solar radiation prediction". en. In: *Neurocomputing* 326-327, pp. 151–160. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2017.05.104. URL: http://www.sciencedirect.com/science/article/pii/S0925231217315229.

Tularam, Gurudeo Anand and Tareq Saeed (May 2016). "Oil-Price Forecasting Based on Various Univariate Time-Series Models". en. In: *American Journal of Operations Research* 6.3. Number: 3 Publisher: Scientific Research Publishing, pp. 226–235. DOI: 10.4236/ajor.2016.63023. URL: http://www.scirp.org/Journal/Paperabs.aspx?paperid=66527.

Udom, P. and N. Phumchusri (2014). "A comparison study between time series model and ARIMA model for sales forecasting of distributor in plastic industry". In: DOI: 10.9790/3021-04213238.

Van Belle, Jente, Tias Guns, and Wouter Verbeke (Jan. 2021). "Using shared sell-through data to forecast wholesaler demand in multi-echelon supply chains". en. In: *European Journal of Operational Research* 288.2, pp. 466–479. ISSN: 0377-2217. DOI: 10.1016/j.ejor.2020.05.059.

Voss, Chris, Nikos Tsikriktsis, and Mark Frohlich (2002). "Case research in operations management". In: *International journal of operations & production management*.

Wang, Qiang, Shuyu Li, and Rongrong Li (Oct. 2018). "Forecasting energy demand in China and India: Using single-linear, hybrid-linear, and non-linear time series forecast techniques". en. In: *Energy* 161, pp. 821–831. ISSN: 0360-5442. DOI: 10.1016/j.energy.2018.07.168. URL: http://www.sciencedirect.com/science/article/pii/S0360544218314658.

Webster, Jane and Richard T. Watson (2002). "Analyzing the Past to Prepare for the Future: Writing a Literature Review". In: *MIS Quarterly* 26.2. Publisher: Management Information Systems Research Center, University of Minnesota, pp. xiii–xxiii. ISSN: 0276-7783. URL: https://www.jstor.org/stable/4132319.

Weytjens, Hans, Enrico Lohmann, and Martin Kleinsteuber (2019). "Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet". In: *Electronic Commerce Research*, pp. 1–21.

William J. Stevenson (2014). *Operations management*. eng. 12th global ed. McGraw-Hill series in operations and decision sciences. Pages: XXXIV, 904. Maidenhead: McGraw-Hill. ISBN: 978-0-07-716952-7.

Xu, Shuojiang, Hing Kai Chan, and Tiantian Zhang (Feb. 2019). "Forecasting the demand of the aviation industry using hybrid time series SARIMA-SVR approach". en. In: *Transportation Research Part E: Logistics and Transportation Review* 122, pp. 169–180. ISSN: 1366-5545. DOI: 10.1016/j.tre.2018.12.005. URL: http://www.sciencedirect.com/science/article/pii/S1366554518308330.

Yenidoğan, I. et al. (Sept. 2018). "Bitcoin Forecasting Using ARIMA and PROPHET". In: *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pp. 621–624. DOI: 10.1109/UBMK.2018.8566476.

Yu, Quan et al. (2018). "Application of Long Short-Term Memory Neural Network to Sales Forecasting in Retail—A Case Study". en. In: *Advanced Manufacturing and Automation VII*. Ed. by Kesheng Wang et al. Lecture Notes in Electrical Engineering. Singapore: Springer, pp. 11–17. ISBN: 978-981-10-5768-7. DOI: 10.1007/978-981-10-5768-7_2.

Zhang, G Peter (2003). "Time series forecasting using a hybrid ARIMA and neural network model". In: *Neurocomputing* 50, pp. 159–175.

Zhang, Guoqiang Peter (2001). "An investigation of neural networks for linear time-series forecasting". In: *Computers & Operations Research* 28.12, pp. 1183–1202.

Zhang, Jie, Gene Lee, and Jingguo Wang (Aug. 2016). "A Comparative Analysis of Univariate Time Series Methods for Estimating and Forecasting Daily Spam in United States". In: *AMCIS 2016 Proceedings*. URL: https://aisel.aisnet.org/amcis2016/ISSec/Presentations/14.

Zhang, Xingyu et al. (Feb. 2014). “Applications and Comparisons of Four Time Series Models in Epidemiological Surveillance Data”. en. In: *PLOS ONE* 9.2. Publisher: Public Library of Science, e88075. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0088075. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0088075.

Appendix A

Additional results

A.1 Plots of training data

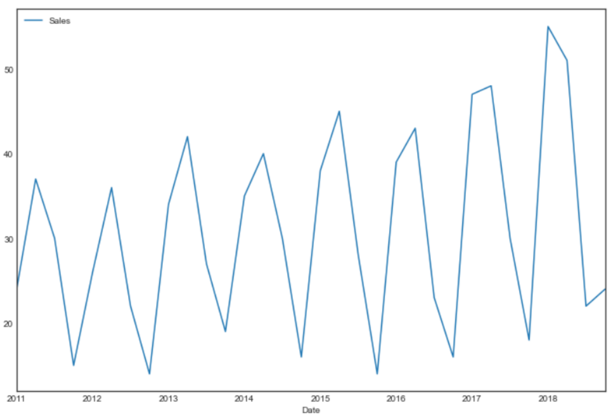

Figure A.1: Graph of actual sales

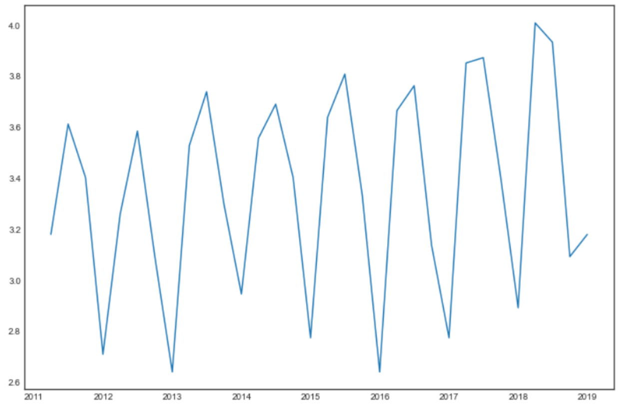

Figure A.2: Graph of log transformed sales

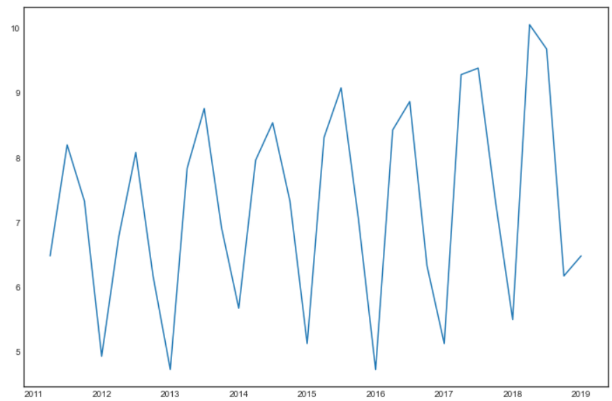

Figure A.3: Graph of Box-Cox transformed sales

A.2 ACF and PACF plots of Log-transformation on training data

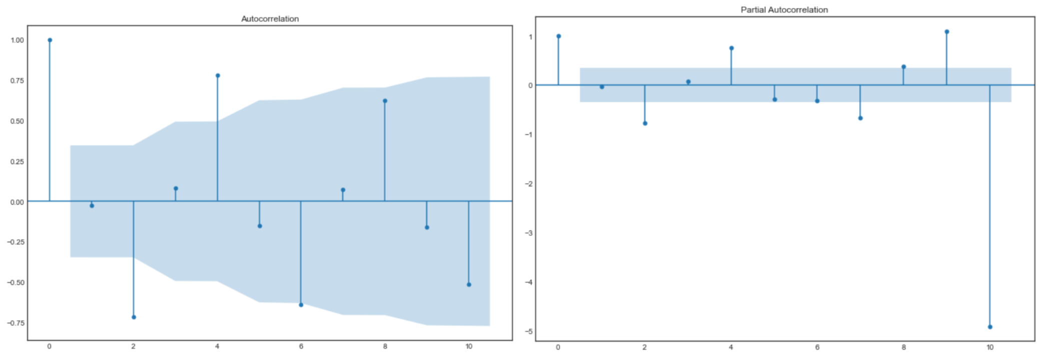

Figure A.4: ACF and PACF of actual training data

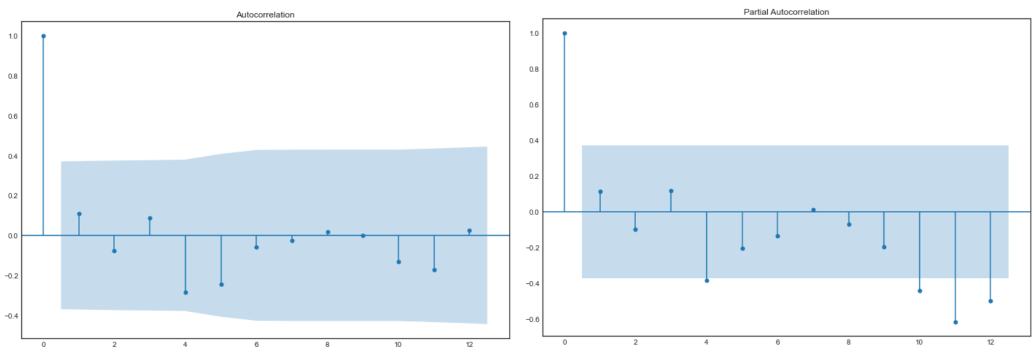

Figure A.5: ACF and PACF plots of seasonally differenced training data

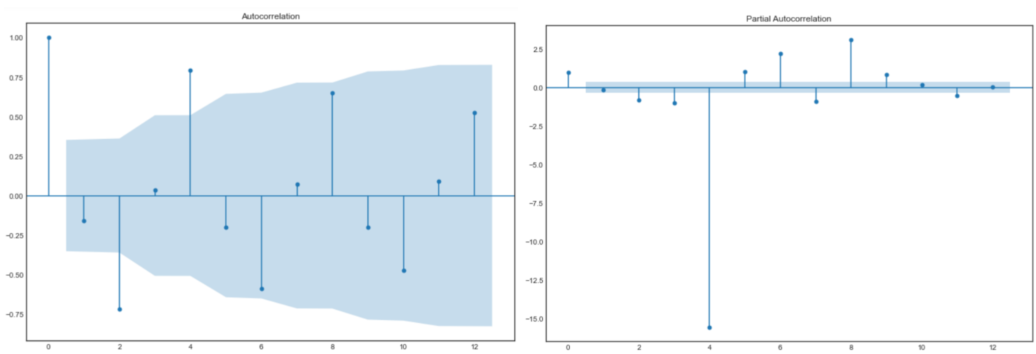

Figure A.6: ACF and PACF plots of first-order differenced training data

A.3 ACF and PACF plots of Box-Cox transformation on training data

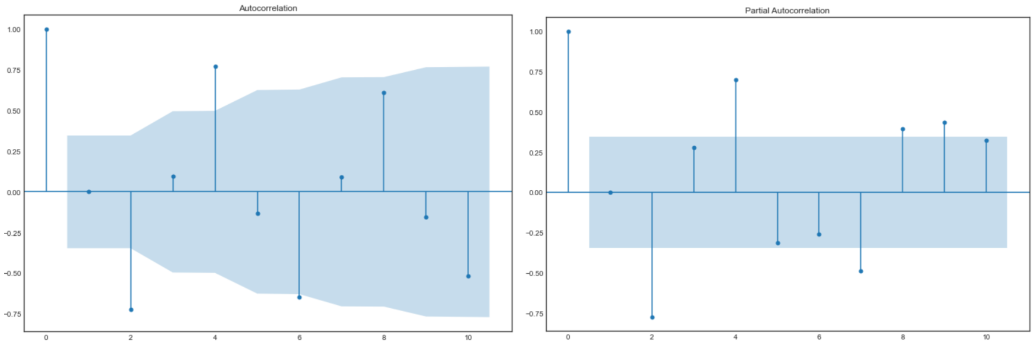

Figure A.7: ACF and PACF of actual training data

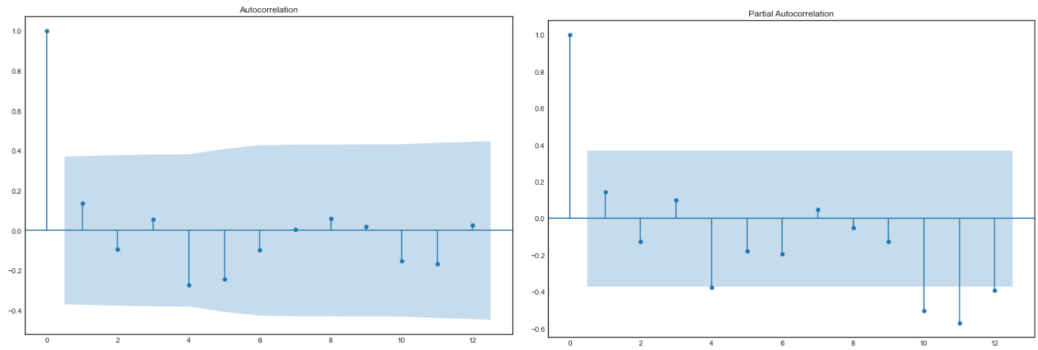

Figure A.8: ACF and PACF of seasonally differenced training data

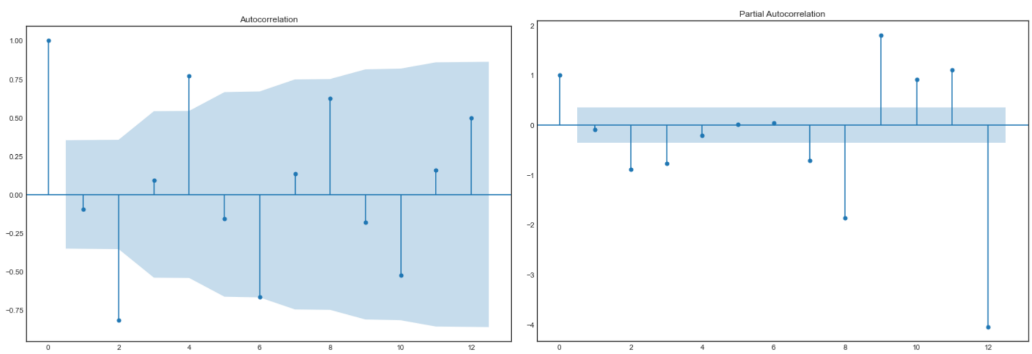

Figure A.9: ACF and PACF of first-order differenced training data

A.4 Residual diagnostics

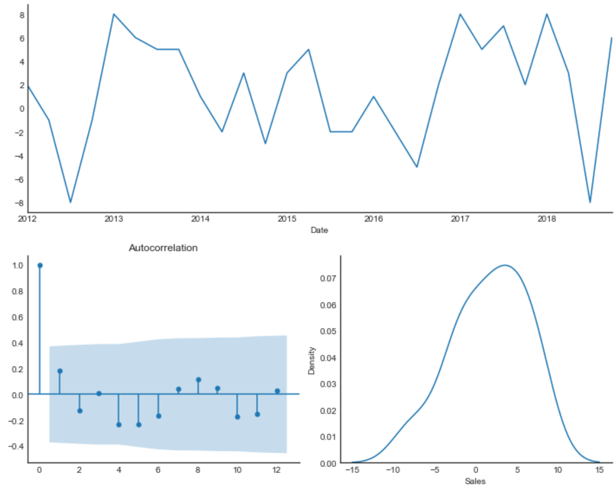

Figure A.10: Residual diagnostics of the Seasonal naive model

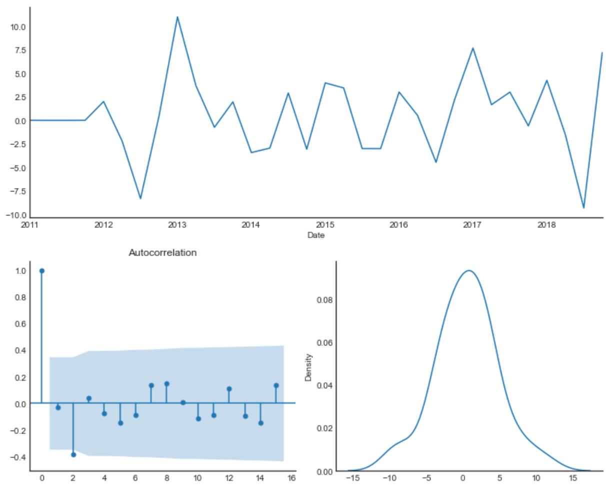

Figure A.11: Residual diagnostics of the Holt-Winters model

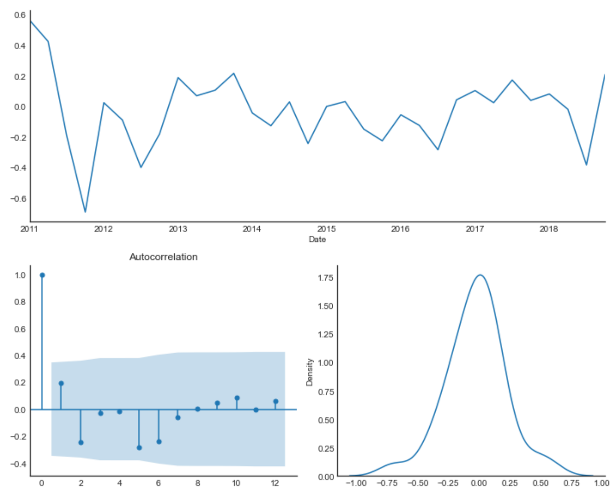

Figure A.12: Residual diagnostics of the Log-ETS(AAA) model

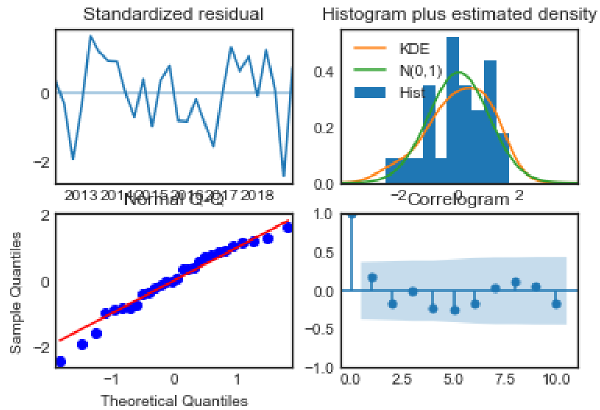

Figure A.13: Residual diagnostics of the SARIMA model with actual series

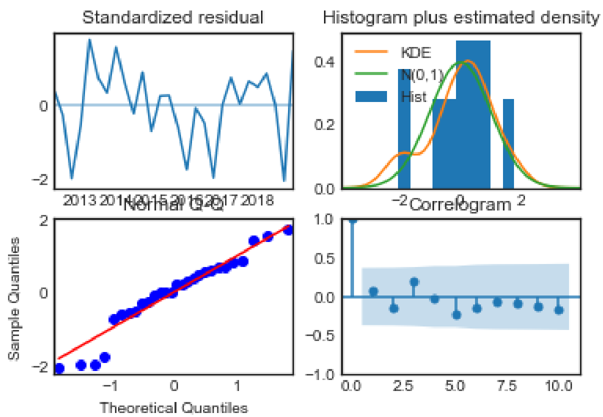

Figure A.14: Residual diagnostics of the SARIMA model with log-transformed series

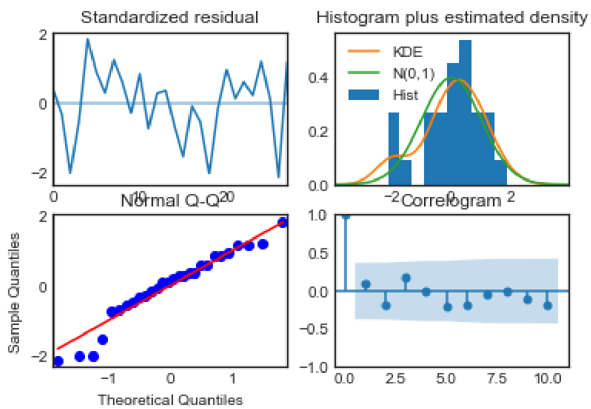

Figure A.15: Residual diagnostics of the SARIMA model with Box-Cox transformed series

Abhilash Ramanathapuram Anand

NTNU
Norwegian University of
Science and Technology