MASTER

Anomaly detection in order to improve the quality of customer forecast data

Zeevenhoven, M.H.J.

Award date:
2020

Link to publication
Anomaly detection in order to improve the quality of customer forecast data

Research paper - Operations Management and Logistics

Author: M.H.J. Zeevenhoven
Supervisor: Dr. ir. R.M. Dijkman
2nd supervisor Dr. ir. W.L. van Jaarsveld
Nexperia Supervisor: Dr. ir. Bas Verheijen
Preface

This master thesis will conclude the master program Operation Management & Logistics at Eindhoven University of Technology and will be used to finalize the graduation project at Nexperia Nijmegen. I would like to use this preface to express my gratitude to a number of people that helped me throughout the graduation project.

First, I would like to thank Bas Verheijen, my supervisor from Nexperia. Without your inspiring presentation at the European Supply Chain Forum, we would have never met. I am thankful for your guidance and support in this project during an unnatural half a year. And our collaborations always inspired me to regard problems from different business angles. Due to the coronavirus, this challenging project turned into an even more challenging one. Working 3 months from home during this time took some adaptation and due to perseverance and great digital support, we made it work.

Secondly, I would like to thank Remco Dijkman as my first supervisor from the university. Due to your critical approach, I learned to clearly structure my project. Your help in the structuring of the research approach as well as in the construction of my report helped me a lot. Additionally, I really enjoyed our digital meetings during this half a year. Thirdly, I would like to thank Willem van Jaarsveld for critically reviewing my progress and helping me with different approaches in order to improve my arguments.

To conclude, this report will be my final deliverable during my masters at Eindhoven University of Technology. I enjoyed the past 6 years with great pleasure, from studying to all side activities that were brought to me during my bachelor and masters degree. This will be the end of a great 6 years, and hopefully, will be the start of something great again. This new start will begin as a supply chain manager for products at Nexperia, who offered me a job after this graduation project. This shows that even in crazy times like a pandemic, great things can happen and personal progress can be made.
Executive Summary

Nexperia is a semiconductor manufacturer that utilizes customer forecast data to predict future demand. Demand managers at Nexperia have experienced anomalies in this customer forecast data. Due to the volume of the data, these anomalies cannot be reviewed manually. Therefore, Nexperia aims to develop an automatic anomaly detection model, that can flag high potential anomalies. The organization does not have extensive decision rules about what anomalies are, and the impact of an anomaly on the planning engine is unknown. Therefore, he model should be able to detect high potential anomalies while the behavior of anomalies is currently unknown. The development of the model is supported by a literature review, and anomalous behavior will be identified by discussions with planning experts from Nexperia. The main research goal of this thesis is:

Design a tool that automatically detects potential anomalies in customer forecast data, in order to improve the data quality.

To reach this goal, an anomaly detection method is constructed, which is evaluated based on the execution within Nexperia.

Method construction

The selected anomaly detection method has three different phases: 'Modeling normal behavior', 'Computing residuals', and 'Classifying observations'. For the modeling of normal behavior two forecasting methods are evaluated: Autoregressive Integrated Moving Average (ARIMA) and Double Exponential Smoothing (DES). To evaluate the performance of both selected anomaly detection methods, labeled training data is collected by consulting planning experts of Nexperia. 212 time series are used as training data, of which 11 time series are anomalous. Each of these 212 time series are divided into 2 sub-time series to take into account the relevance difference between forecasts for the first 13 weeks, and the period after the first 13 weeks.

Classification performance

Tables 1 & 2 show the classification performance for different combinations of classification boundaries for the sub-time series.

Table 1: Classification results DES forecasting

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Acc</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>19</td>
<td>2</td>
<td>182</td>
<td>0.90</td>
<td>0.32</td>
<td>0.82</td>
<td>0.46</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>3</td>
<td>193</td>
<td>0.95</td>
<td>0.50</td>
<td>0.73</td>
<td>0.59</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>193</td>
<td>0.94</td>
<td>0.47</td>
<td>0.64</td>
<td>0.54</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>199</td>
<td>0.97</td>
<td>0.75</td>
<td>0.55</td>
<td>0.63</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>199</td>
<td>0.96</td>
<td>0.71</td>
<td>0.45</td>
<td>0.56</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>201</td>
<td>0.96</td>
<td>1.00</td>
<td>0.27</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table 2: Classification results ARIMA forecasting

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Acc</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>24</td>
<td>2</td>
<td>177</td>
<td>0.88</td>
<td>0.27</td>
<td>0.82</td>
<td>0.41</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>13</td>
<td>3</td>
<td>188</td>
<td>0.92</td>
<td>0.38</td>
<td>0.73</td>
<td>0.50</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>13</td>
<td>5</td>
<td>188</td>
<td>0.92</td>
<td>0.32</td>
<td>0.55</td>
<td>0.40</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>200</td>
<td>0.98</td>
<td>0.88</td>
<td>0.64</td>
<td>0.74</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>200</td>
<td>0.97</td>
<td>0.83</td>
<td>0.45</td>
<td>0.59</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>201</td>
<td>0.96</td>
<td>1.00</td>
<td>0.27</td>
<td>0.43</td>
</tr>
</tbody>
</table>
The tables show that both anomaly detection methods achieve the highest performance when both sub-time series utilize a classification boundary of 2 interquartile ranges on the boxplot. The interquartile ranges are computed based on historic behavior of the time series. ARIMA forecasting shows to be outperforming DES forecasting based on the $F_1$ measure. However, the computation time of both methods are shown in Table 3. This shows that DES outperforms ARIMA by 6 times if computational efficiency would be taken into account.

Table 3: Computational efficiency in seconds when executing the grid search and execution process

<table>
<thead>
<tr>
<th>#time series</th>
<th>Grid Search (sec)</th>
<th>Execution (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARIMA</td>
<td>DES</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARIMA</td>
</tr>
<tr>
<td>424</td>
<td>424</td>
<td>4500</td>
</tr>
<tr>
<td>1</td>
<td>10.6</td>
<td>1.6</td>
</tr>
<tr>
<td>30000</td>
<td>318396 (88+ hrs)</td>
<td>48891 (13+ hrs)</td>
</tr>
</tbody>
</table>

The comparison results in a trade-off between classification performance and computational efficiency. This trade-off should be taken into account when selecting the optimal method.

Implications

The results of the initial execution, produced several implications. First, an error rate of 5% would be impractical, which would imply a weekly review of 800+ potential anomalies. Secondly, the used training data was collected from three customers, however, Nexperia has hundreds of customers in different industries. Therefore, this raised generalizability implications, if the method would be executed for different customers. Finally, the impact of an anomaly on the production planning is unknown, which results in difficulties with expressing the value of an anomaly detection model. These implications were evaluated in a feasibility study, and solution directions were proposed.

Feasibility study

The error rate in the training data was assessed to be too high, therefore, distinction has been made between major and minor errors. By only reviewing the major errors, which are most relevant for Nexperia, the following performance of Table 4 was achieved:

Table 4: Performance when excluding the minor errors anomalies

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>$F_1$ score DES major anomalies</th>
<th>$F_1$ score ARIMA major anomalies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.24</td>
<td>0.21</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.38</td>
<td>0.31</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.3</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0.62</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.62</td>
<td>0.55</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

As expected, when only detecting major anomalies, the optimal classification boundaries increase. Both methods show similar performance when only selecting major anomalies, which indicates that DES anomaly detection outperforms ARIMA anomaly detection, because of the faster computation time. The error rate of solely major anomalies is still 300+ anomalies on a weekly basis, which is infeasible for Nexperia. Therefore, we recommend to take into account demand volume of the anomalies. The demand volume of an anomaly can highly influence its relevance. If an anomalous forecast is received which is a minor part of the total demand for a product, the impact of such an anomaly on the production planning is negligible. However, if an anomaly represents a large part of the total demand of a product, this directly can have impact on the ability to fulfill demand. Therefore, demand volume of an anomaly
has to be taken into account when developing such an anomaly detection model. This is not implemented in this thesis because of the inability to combine this data. However, several relevant insights about the demand volume of anomalies are discussed in the thesis. Finally, the generalizability between different customers is evaluated. A test set of 53 time series is extracted from different customers and evaluated based on the distribution of error scores and number of anomalies. Table 5 shows the number of potential anomalies raised for the different classification boundary combinations. This table indicates an increasing similarity in detection rate when the classification boundaries increase. This again shows that the most important anomalies are located above the 2 * interquartile range of the boxplot. Additionally, this shows potential that the anomaly detection methods are generalizable for customers from different industries.

Table 5: Comparison error distribution test and training dataset

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound 13+</th>
<th>Training anomaly rate DES</th>
<th>Training anomaly rate ARIMA</th>
<th>Test anomaly rate DES</th>
<th>Test anomaly rate ARIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.13</td>
<td>0.16</td>
<td>0.21</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.08</td>
<td>0.10</td>
<td>0.15</td>
<td>0.19</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.07</td>
<td>0.09</td>
<td>0.13</td>
<td>0.19</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.04</td>
</tr>
</tbody>
</table>

**Recommendations**

The anomaly detection methods show large potential when evaluating major anomalies. However, the explicit value of correcting an anomaly is hard to determine. Therefore, the major anomalies should be combined with the demand volume to evaluate the impact of an anomaly on the production planning. Additionally, this ensures that the anomalies that can lead to the inability to fulfill demand are removed. Several design choices are introduced and it should be noted that these design choices are selected based on the used training data. Before implementing the proposed anomaly detection method, these design choices should be evaluated for a larger and more diverse dataset. Finally, the knowledge about anomalous behavior would be extremely beneficial for an anomaly detection method. Currently, limited simplistic methods are used to evaluate the customer forecast data. However, if examples of anomalous behavior were present (labeled data), a more comprehensive detection model could be developed by introducing machine learning techniques.
Abstract

In order to evaluate the quality of customer forecast data, anomaly detection applications are widely used. This thesis designs the most effective anomaly detection model in order to evaluate the data quality at a leading semiconductor manufacturer. First, we conducted a literature review to identify effective anomaly detection methods for time series. Additionally, evaluation measures were identified in this literature review. Secondly, two different forecasting based detection methods are compared. Our evaluation shows that anomalies can be detected with a high accuracy by both forecasting methods. The ARIMA method achieved a $F_1$ score of 0.74, in comparison with a $F_1$ score of 0.63 for Double Exponential Smoothing. However, Double Exponential Smoothing outperforms Autoregressive Integrated Moving Average, based on computational efficiency.

Finally, the feasibility of the method at the semiconductor manufacturer is evaluated by conducting a feasibility study. This study showed three key findings: first, the method showed similar error scores on a test dataset which supports the generalizability of the method for different customer industries. Secondly, implications were identified about the total anomaly volume flagged on a weekly basis. To account for these implications, minor anomalies were excluded, which resulted in a more approachable anomaly volume. Additionally, this exclusion increased the classification performance for both forecasting methods to a $F_1$ score of 0.75. Thirdly, the relevance of the method could be increased by taking into account the demand volumes of flagged anomalies. Based on the performance of the model and the findings from the feasibility study, we can conclude that the introduced model would be effective for Nexperia.
# List of Figures

1.1 A visualization of the planning application of Nexperia ................................. 2  
1.2 The planning horizon of Nexperia .............................................................. 3  
1.3 A visualization of the thesis methodology .................................................. 6  

2.1 An imaginary EDI received by Nexperia ......................................................... 10  
2.2 A cumulative time series construction, where the construction for two subsequent observations are visualized ................................................................. 12  
2.3 A visualization an EDI forecast ................................................................. 14  

5.1 A simplified modeling of computed normal behavior for this time series .......... 26  
5.2 Residuals computed from Figure 5.1 ......................................................... 27  
5.3 To be classified observation with corresponding classification boundaries .......... 28  
5.4 Construction of a boxplot ...................................................................... 30  

6.1 The decreasing relevance of historical observations, when trends are present .... 32  
6.2 The relevance of using different intervals to construct the cumulative time series ... 34  
6.3 A visualization of a cumulative time series, with a length of 39 observations. .... 35  
6.4 The cumulative time series of Figure 6.3 divided into two sub-time series. ...... 35  
6.5 A visualization of the anomaly detection model in the organization of Nexperia .... 38  
6.6 The transformation of the classification boundaries if the decision rules are satisfied. ... 40  
6.7 IQR distances from week 1-13 with the exponential smoothing method ........ 41  
6.8 IQR distances from week 13+ with the exponential smoothing method .......... 42  
6.9 IQR distances from week 1-13 with the ARIMA method ............................... 43  
6.10 IQR distances from week 13+ with the ARIMA method .............................. 43  
6.11 Histogram of ARIMA forecasting error scores for 1-13 weeks (Training) ......... 45  
6.12 Histogram of ARIMA forecasting error scores for 1-13 weeks (Test) ............. 45  
6.13 Histogram of ARIMA forecasting error scores for 13+ weeks (Training) .......... 45  
6.14 Histogram of ARIMA forecasting error scores for 13+ weeks (Test) ............. 45  
6.15 Histogram of DES forecasting error scores for 1-13 weeks (Training) ............ 46  
6.16 Histogram of DES forecasting error scores for 1-13 weeks (Test) ................. 46  
6.17 Histogram of DES forecasting error scores for 1-13 weeks (Training) .......... 46  
6.18 Histogram of DES forecasting error scores for 13+ weeks (Test) ................. 46  

A.1 A visualization of an EDI made in week 18 of 2019 .................................. 58  
A.2 A visualization of an EDI made in week 19 of 2019 .................................. 58  
A.3 A visualization of an EDI made in week 20 of 2019 .................................. 59  

E.1 A example dashboard including the cumulative time series as the EDI forecasts. ... 64
List of Tables

1. Classification results DES forecasting ........................................ ii
2. Classification results ARIMA forecasting ..................................... ii
3. Computational efficiency in seconds when executing the grid search and execution process iii
4. Performance when excluding the minor errors anomalies .................. iii
5. Comparison error distribution test and training dataset .................... iv

1.1. Goals of the research questions, accompanied with the research methods and deliverables. 8

2.1. Overview of data characteristics ............................................ 13

3.1. Anomaly detection methods list ............................................. 16
3.2. Anomaly detection methods with corresponding exclusion reasons .......... 20

5.1. Pseudo code anomaly detection ............................................. 31

6.1. Stationary Dickey-Fuller test ................................................ 33
6.2. Results exponential smoothing ............................................. 42
6.3. Results ARIMA .............................................................. 44
6.4. Comparison error distribution test and training dataset .................. 45
6.5. Computational efficiency in seconds when executing the grid search and execution process 47
6.6. Performance when excluding the minor errors anomalies .................. 47

B.1. Exclusion and Inclusion of variables in the training dataset .............. 60
# List of abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIC</td>
<td>Akaike Information Criterion</td>
</tr>
<tr>
<td>ARIMA</td>
<td>Autoregressive Integrated Moving Average</td>
</tr>
<tr>
<td>AUC</td>
<td>Area Under Curve</td>
</tr>
<tr>
<td>BIC</td>
<td>Bayesian Information Criterion</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Demand Forecast</td>
</tr>
<tr>
<td>CPN</td>
<td>Customer Part Number</td>
</tr>
<tr>
<td>DES</td>
<td>Double Exponential Smoothing</td>
</tr>
<tr>
<td>EDI</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>EM</td>
<td>Evaluation Measure</td>
</tr>
<tr>
<td>FN</td>
<td>False Negative</td>
</tr>
<tr>
<td>FP</td>
<td>False Positive</td>
</tr>
<tr>
<td>IQR</td>
<td>Interquartile Range</td>
</tr>
<tr>
<td>MAE</td>
<td>Mean Absolute Error</td>
</tr>
<tr>
<td>MAPE</td>
<td>Mean Absolute Percentage Error</td>
</tr>
<tr>
<td>ML</td>
<td>Machine Learning</td>
</tr>
<tr>
<td>QTY</td>
<td>Quantity</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Squared Error</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>TN</td>
<td>True Negative</td>
</tr>
<tr>
<td>TP</td>
<td>True Positive</td>
</tr>
</tbody>
</table>
# Table of contents

## 1 Introduction

1.1 Organizational description ............................................. 1
   1.1.1 Organizational structure ........................................ 1
   1.1.2 Planning Horizon .................................................. 2
   1.1.3 Problem description ............................................... 3

1.2 Research design ....................................................... 4

1.3 Research methodology and deliverables ............................... 5
   1.3.1 Anomaly detection methods ....................................... 5
   1.3.2 Performance evaluation and improvement .......................... 5
   1.3.3 Support activities ................................................. 7
   1.3.4 General research goal ............................................. 7

1.4 Contribution to literature ............................................. 8

1.5 Report structure ...................................................... 9

## 2 Data Understanding and Preparation ................................. 10

2.1 Customer forecast data exchange ..................................... 10

2.2 Construction of cumulative time series .............................. 11
   2.2.1 Concept evaluation ............................................... 12

2.3 Data preprocessing .................................................... 13
   2.3.1 Data characteristics ............................................. 13
   2.3.2 Cumulative construction ......................................... 14
   2.3.3 Data exclusion ................................................... 14

2.4 Summary ............................................................ 15

## 3 Method Selection ...................................................... 16

3.1 Anomaly Detection methods ......................................... 16
   3.1.1 Machine Learning methods ....................................... 17
   3.1.2 Forecasting based classification ................................ 17
   3.1.3 Forecasting classification process ............................... 18

3.2 Method selection .................................................... 19
# Table of contents

3.2.1 Organizational restrictions ........................................ 19  
3.2.2 Selection procedure ............................................. 19  
3.3 Conclusion ......................................................... 21  

4 Evaluation technique selection .................................... 22  
4.1 Forecast evaluation measures ....................................... 22  
4.1.1 AIC/BIC ......................................................... 22  
4.1.2 Mean Absolute Percentage Error (MAPE) ..................... 22  
4.1.3 Root Mean Squared Error (RMSE) ............................. 22  
4.1.4 Mean Absolute Error (MAE) .................................... 23  
4.2 Classification Evaluation ........................................... 23  
4.2.1 Confusion matrix measures ...................................... 23  
4.3 Conclusion ......................................................... 24  

5 Anomaly detection execution ........................................ 26  
5.1 Execution overview ................................................ 26  
5.2 Modeling normal behavior .......................................... 28  
5.2.1 Autoregressive Integrated Moving Average (ARIMA) .... 28  
5.2.2 Double exponential smoothing ................................... 29  
5.3 Anomaly classification .............................................. 30  
5.3.1 Boxplot ......................................................... 30  
5.4 Pseudo algorithm ................................................... 31  

6 Implementation and Results ......................................... 32  
6.1 Design choices ...................................................... 32  
6.1.1 Time-dependent relevance - Normal behavior horizon .... 32  
6.1.2 Relevance difference between forecast in the near and far future ................................................... 33  
6.1.3 Classification boundaries ....................................... 35  
6.2 Forecasting parameter optimization ................................ 36  
6.2.1 Grid search ...................................................... 36
# Table of contents

6.2.2 Concept drift ........................................... 37  
6.3 Evaluation at Nexperia .................................. 37  
6.3.1 Management control ................................. 38  
6.3.2 Employee Feedback ................................... 39  
6.4 Results ..................................................... 40  
6.4.1 Training data .......................................... 40  
6.4.2 Classification performance ....................... 41  
6.5 Feasibility & validation ................................ 44  
6.5.1 Validation of generalizability .................... 44  
6.5.2 Computational efficiency ......................... 46  
6.5.3 Anomaly volume ..................................... 47  
6.6 Discussion ................................................... 48  

7 Conclusion & recommendations ............................ 49  
7.1 Research goal and questions ......................... 49  
7.2 Limitations .................................................. 51  
7.2.1 Impact on the planning engine ................... 51  
7.2.2 Dataset limitations .................................. 51  
7.2.3 Cumulative series limitations .................... 51  
7.2.4 Employee feedback .................................. 52  
7.3 Future research ............................................. 52  
7.4 Recommendations ......................................... 53  
7.4.1 Design choices ........................................ 53  
7.4.2 feasibility major and minor anomalies .......... 53  
7.4.3 Employee review ...................................... 53  
7.4.4 Demand volume of anomalies .................... 54  
7.4.5 Collection of anomalies ........................... 54  
7.4.6 Dashboard .............................................. 54  

Appendix A ....................................................... 58
<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix B</td>
<td>Appendix B</td>
<td>60</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Appendix C</td>
<td>61</td>
</tr>
<tr>
<td>C.1</td>
<td>ARIMA explanation</td>
<td>61</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Appendix D</td>
<td>63</td>
</tr>
<tr>
<td>Appendix E</td>
<td>Appendix E</td>
<td>64</td>
</tr>
</tbody>
</table>
1 Introduction

This thesis is the result of a research conducted at a leading semiconductor manufacturer. The thesis focuses on evaluating forecasting data received from downstream customers. Customer forecast data can be used to make predictions about the future demand volume. This information is extremely valuable for suppliers in order to manage and optimize production operations. Inaccurate forecasts could lead to product surplus or shortage at the supplier. Therefore, accurate data exchange in supply chains is essential to optimize the overall performance for both customer and supplier. This thesis focuses on ensuring and improving data quality in supply chains by developing an anomaly detection model. An anomaly detection model can be used to detect errors and unexpected events in datasets and will be used to improve the customer forecast data quality.

The remainder of this chapter introduces the organization and its relevant operational processes in section 1.1. This section includes a problem description, where the experienced problems are discussed. This problem description is translated to a research goal, for which research questions are derived in section 1.2. To answer these research questions, a research methodology is visualized and discussed in section 1.3. After introducing the research methodology, the contribution of this research to literature is discussed in section 1.4. Finally, the remaining structure of this thesis is discussed in 1.5.

1.1 Organizational description

This research is conducted at Nexperia, a global leader in the manufacturing of semiconductors. This section describes the organization and their operational activities related to the demand planning, for which customer forecast data is used. In subsection 1.1.1 the organizational structure is discussed and visualized to give a clear perception of the organization. The relevant business processes for this research are discussed and elaborated in more detail. This elaboration will be followed by section 1.1.2, where the planning horizon is shown to zoom more into the relevance of different planning horizons. Finally, the experienced problems are discussed in 1.1.3.

1.1.1 Organizational structure

The production process of these semiconductors, is divided into front-end activities, and back-end activities. The front-end factories, located throughout Europe, produce wafers on which products are patterned. Afterwards, the front-end end products are transferred to the back-end factories located throughout Asia. In these back-end factories the products are extracted from the wafers, modified for customer needs, and eventually packaged. All of these activities are executed by machines with various characteristics, such as capacities, lead times, and yield. To optimally manage the production planning while taking into account these characteristics, a planning application is used.

This planning application is driven by a linear program that takes into account several inputs and restrictions. Input for the planning engine is the customer demand, which is restricted by the capacity, yield and lead times of the machines. Customer demand is divided into sales orders and demand forecasts. The sales orders are tangible orders that customers placed with a deadline and the demand forecasts are the expected demand volumes needed in the future. These demand forecasts are collected from customers and are used to optimize the production planning. The linear program is executed by the i2 demand planning software and is executed on a weekly basis at Nexperia [40]. The goal of the planning engine is to minimize the lateness of the planned sales orders. The planning application with regard to the organizational structure is visualized in Figure 1.1.
To take into account differences between demand types, Nexperia utilizes objective rules for their linear program. Demand in the planning engine is allocated a priority level, based on aspects such as demand type, contractual agreement, revenue from customer, relationship, customer lead time etc.. The planning application can take into account these priorities and first account for demand with higher priorities.

Additionally, differences in forecasting behavior exist. Some customers do share their demand forecasts, which internally is known as Electronic Data Interchange (EDI). Other customers do not share their demand forecasts, in which case Nexperia forecasts future demand for these customers. For the purpose of this research are generalizing both forecasting approaches. Forecasts made by customers, and forecasts made by Nexperia are both be evaluated. It is assumed that 15.000 forecasts enter the planning engine on a weekly basis, and should all be dealt with. In the remainder of this thesis we use the term customer demand forecast for both forecasting approaches.

The linear program is executed on a weekly basis and two different planning schematics will be produced: the requirement scenario and the capacity scenario. The requirement scenario fulfills all desired demand for the coming planning windows, without taking into account capacity constraints. The capacity scenario introduces the operational characteristics such as capacity and expected yield. The requirement scenario focuses on the projected demand for the coming planning window, while the capacity scenario focuses on the ability to fulfill that demand. The differences between these scenarios are analysed and based on these differences, decisions are made for the future regarding planning and capacities.

### 1.1.2 Planning Horizon

To produce 90+ billion products yearly with lead times of 15 weeks, production planning is extremely important. To ensure a high accuracy for the planning application, customer EDI demand forecast is used in addition to sales orders. Together this represents the planning horizon. Figure 1.2 shows a visualization of the production planning for the coming 6 quarters. Blue bars indicate the EDI, which often is supplemented by internal forecasts by Nexperia, because most customers only forecast up to 18-52 weeks. Orange bars indicate the received tangible sales orders for that quarter. Red bars show the already fulfilled sales orders.

![Figure 1.1: A visualization of the planning application of Nexperia](image-url)
Currently, Nexperia already executes anomaly detection to ensure large errors do not enter the planning engine. This anomaly detection is executed solely for the forecasts for the coming 13 weeks. If the total forecasted demand for the coming 13 weeks changes with more than 33% the forecasts are flagged and checked by planning employees. This interval of 13 weeks is taken because this has the most impact on the ability to fulfill future demand. Moreover, Nexperia assumes that future forecasts decrease in relevance over time, because anomalies in the distant future could be accounted for later. Although the difference in relevance, forecasts for the period after the first 13 weeks, could have an impact on the ability to fulfill demand as well. Therefore, the total planning horizon should be evaluated to identify anomalies and their relevance.

### 1.1.3 Problem description

With the increasing use of data in organizations, the advantages of high quality data are increasing concurrently [31]. High quality data can increase the value of data immensely. However, acquiring high quality data is resource intensive and therefore expensive to gather. The volume of data that Nexperia handles requires an efficient method to ensure the quality of the data. Anomaly detection methods are increasing in popularity to fill this gap between the desire to improve data quality and the high cost accompanied with improving the quality [5]. Nexperia utilizes a planning engine in which sales orders, and demand forecasts complement each other. However, the demand managers have experienced incidents where incorrect demand forecasts have entered the planning engine.

The issues with evaluating these demand forecasts are mainly caused by the volume of data. 15,000 demand forecasts enter the planning engine on a weekly basis. This volume is too big to be evaluated manually by employees. Currently, only the customer demand forecasts for the next 13 weeks are evaluated by a simplified changepoint detection. This detection flags forecasts of which the total forecasted demand changed with more than 33%. This detection is extremely simplified and does not take into account the variability, trend or seasonality in the past, while only focusing on the next 13 weeks of the planning horizon. Therefore, the organization expressed the need for an automatic method of detecting anomalies in this customer forecast data. Furthermore, the forecasts beyond the first 13 weeks are interesting for future purchase decisions, based on the capacity scenario described in subsection 1.1.2. Therefore, a more comprehensive tool should be introduced, able to automatically detect anomalies in time series with a horizon longer than 13 weeks.
The main causes of these incidents are:

- **employee & customer errors**

  While working with large volumes of data, human errors always exist [15]. This can be mistakes due to inexperience, or simple input errors. However, these errors can have a large impact on the planning engine of Nexperia, when unaccounted for. These errors can lead to the inability to fulfill demand, or excessive inventory.

- **Planning nervousness**

  In variable markets, customers can share exaggerated forecasts to make sure Nexperia always can deliver fluctuating demand [28]. However, this can be extremely harmful for Nexperia as a supplier when the customer demand is significantly lower than forecasted in the past. This could lead to surplus in production or shortage in inventory.

  Planning nervousness leads to a Bullwhip effect in the total supply chain. The Bullwhip effect states that, going further down the supply chain, the variability increases. This shows that inaccurate forecasts would not only impact Nexperia, but also its customers. Utilizing accurate information in earlier stages of the supply chain can decrease the Bullwhip effect throughout the entire supply chain [31]. Decreasing this so-called 'Bullwhip effect' leads to a decrease in production and inventory costs.

Multiple studies have observed a causal relation between inaccurate forecasts and the bullwhip effect [28], [31], [33]. Therefore, to improve operational performance at the semiconductor manufacturer, inaccurate demand forecasts should be investigated and corrected where needed.

Based on the desire to introduce an automatic anomaly detection method, two main problems arise: which method to use, and optimizing its performance. These issues will be discussed in their respective chapters throughout the thesis and will be translated to tangible research questions in the next section.

### 1.2 Research design

The following research goal is derived from the problem description above:

*Design a tool that automatically detects potential anomalies in customer forecast data, in order to improve the data quality.*

Given the research goal, the following research questions are formulated:

1. **What current anomaly detection methods exist that are suitable for time series data?**

2. **How do the selected anomaly detection methods perform in practise, and how to optimize this performance?**

**Scope and assumptions**

To ensure that the research is approachable and executable, the following scoping decisions and assumptions are introduced.

- We will utilize three different customers of Nexperia for the training of the model. These customers exist in different industries to give a comprehensive overview, while keeping the computation time approachable (262 different product-customer combinations will be evaluated).

- To test the model two additional customers are used to validate the model (53 time series).
• Customer forecast data for the coming 13 weeks is the most impactful. However, forecast data beyond the coming 13 weeks could also impact the planning engine.

• The time series do not include seasonality.

• The available data has no accompanied classification about whether or not it is anomalous.

• The customer forecast data is received on a weekly basis and should be evaluated on a weekly basis.

For the execution of this research framework, several preconditions have to be met to keep the research executable:

• The demand is the single explanatory variable (univariate) in the time series, and should be present for all received forecasts.

• Historical data is available for all products.

• Employees have knowledge about when an observation is an anomaly, and employees always make a correct predictions in the review process.

### 1.3 Research methodology and deliverables

To analyse the problem situation, a structured methodology will be used. This methodology includes the research questions and combines the answers of the research questions to give a comprehensive solution to the problem. The methodology is visualized in Figure 1.3, and is discussed in the following subsections. Activities are visualized by rounded rectangles, and deliverables with straight rectangles. The colours represent the different research questions, or support activities, each of which has a separate chapter in the remainder of this thesis. The chapter in which the activity is executed is visualized by the numbers accompanied in the visualization of Figure 1.3.

#### 1.3.1 Anomaly detection methods

The first research question focuses on the anomaly detection method selection. This process contains a literature review, where multiple research papers are extracted and analysed in order to present an effective anomaly detection method relevant for our problem situation. The selection process is executed based on requirements and limitations of our problem situation. This process is supported by several support activities which will be discussed later.

15 detection methods result from the literature review and the most effective method will be selected. This method includes a forecasting process, of which two alternatives are implemented. The results of both forecasting methods in relation to the final performance are evaluated in the following methodology steps.

#### 1.3.2 Performance evaluation and improvement

The second research question focuses on the evaluation and improvement of the anomaly detection method. Similar to the first research question, the evaluation measures are extracted via a literature review. This literature review identifies evaluation measures for the forecasting process as well as the classification process. The most effective methods will be selected and used for the implementation. Labeled data is used to evaluate the performance of the model. This labeling is executed by a support activity. Afterwards, this labeled data is used to evaluate the performance of the anomaly detection model. This evaluation is used to identify strengths and weaknesses of the model.
Figure 1.3: A visualization of the thesis methodology
1.3.3 Support activities

The research goal is to develop a tool that automatically detects potential anomalies in forecast data. This identifies two necessities for the methodology, the development of a tool and the evaluation of forecast data. This forecast data should be analysed and interpreted in order to preprocess the data. Understanding data, and its use in organizations is crucial for the development of data analytic models [39]. Therefore, the following support activities are introduced: 'Understand and preprocess the forecast data', 'aggregate/manipulate the data’, and 'label the data’. All data manipulations are executed in Python 3.7.

First, in order to understand the value of the data, the forecast data should be explored. This exploration should take into account that the data can be used to achieve the research goal. Secondly, the preprocessing should remove all unnecessary characteristics and determine a relevant format for the data. Finally, it is possible that the value of the data is not directly present, and that the data should be manipulated or aggregated to target the research goal. These three steps should ensure that the data has a format that is useful, and that the final dataset contains value towards the main research goal.

As discussed, to evaluate the performance of the anomaly detection model, labeled data is necessary to give a comprehensive evaluation. This collection of labeled training data is executed by consulting planning experts. These experts are asked to identify anomalies in the preprocessed dataset. Afterwards, this data is used to compare the forecasting methods in relation to the model performance.

1.3.4 General research goal

Finally, the activities for the first and second research question as well as the support activities will be used as input for the model development. The most effective detection methods, valuable aggregated data, and evaluation techniques will be used for the implementation. The anomaly detection method will be built in Python, where the labeled forecasting data will be used to train the anomaly detection model. The classification results will be evaluated in order to evaluate the initial performance. Finally, the evaluation is used to select the best method design and implement the method for Nexperia. By designing such a method, several design choices have to be made, and the feasibility of the method should be taken into account. Therefore, a feasibility study will be executed to identify whether the presented model is relevant for the organization. Aspects such as generalizability, executability and computational efficiency will be discussed. Finally, conclusions will be made about the entire anomaly detection model, and limitations will be addressed. Based on these conclusions and limitations, recommendations are discussed for Nexperia, to provide guidance and insight, in order to implement the anomaly detection method into the organization.

To summarize, table 1.1 contains the research questions accompanied by the respective goals, activities and results.
Table 1.1: Goals of the research questions, accompanied with the research methods and deliverables.

<table>
<thead>
<tr>
<th>Research question</th>
<th>Goal</th>
<th>Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify effective anomaly detection methods</td>
<td>literature review</td>
<td>2 best performing methods selected from 15 anomaly detection methods</td>
</tr>
<tr>
<td>2</td>
<td>Evaluate the performance of the methods resulting from research question 1</td>
<td>Identify anomalies in the training data by consulting planning experts</td>
<td>Major and minor anomalies in 212 time series</td>
</tr>
<tr>
<td></td>
<td>Literature review</td>
<td>Relevant evaluation measures</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluate forecasting performance</td>
<td>Mean Absolute Error for 212 time series</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Evaluate classification performance</td>
<td>$F_1$ score for 212 time series</td>
<td></td>
</tr>
<tr>
<td>Main</td>
<td>Develop anomaly detection tool to evaluate customer forecast data quality</td>
<td>Implement the best performing method from research question 2</td>
<td>Anomaly detection method able to identify anomalies with high accuracy for the used dataset</td>
</tr>
<tr>
<td></td>
<td>Execute feasibility and validation study to identify whether the method is effective for Nexperia</td>
<td>Assessment about the generalizability and applicability of the method in the organization</td>
<td></td>
</tr>
</tbody>
</table>

1.4 Contribution to literature

As can be derived from the research goal, the goal is to develop an anomaly detection model that can automatically detect anomalies in customer forecast data. This customer forecast data can be modeled as univariate time series with demand as sole variable. From the initial literature scan we derived three general types of anomaly detection applications applicable to univariate time series exist in the literature. The first type consists of Machine Learning methods. These methods use high volumes of training data with labeled anomalies to learn the characteristics of anomalies. This is often regarded as a very accurate and computational efficient method. However, many high quality labeled training data is necessary, which is often unavailable in organizations [7].

The second type of anomaly detection methods is the detection of intervals. In this situation, total intervals are classified as anomalous or not based on large variability or disruptive behavior in this interval. Statistical characteristics for the entire interval are computed and disruptive behavior results in the classification of the entire interval as an anomaly.

In our problem situation only new weekly observations should be evaluated. Additionally, no labeled training data exists inside the organization. Therefore, this thesis will implement the third category of anomaly detection methods: forecasting based classification. These methods use forecasting methods to model normal behavior and detect anomalies based on the (dis)similarity to this forecasted behavior. In literature this concept of forecasting normal behavior is applied multiple times, however few of these cases are based on demand forecasting in supply chains [27]. Additionally, these research papers purely focused on the classification, while neglecting the impact of a method on the organization. This thesis will focus on design choices for Nexperia as well as the impact of such a method on the production planning. The main design choice that is introduced focuses on the relevance difference between forecast...
in the near and far future. In addition, the presence of anomalies in the training data can impact the modeling of normal behavior. Few authors in academic literature had to deal with unknown anomalies in the training data, when modeling normal behavior. Finally, the classification of the anomalies is often done by parametric approaches. However, because of the existing anomalies in the training data, a non-parametric approach will be taken in order to fill this gap in the literature.

By filling these gaps, anomaly detection methods could be developed that are robust for future changes, while anomalies in the training data would be allowed.

1.5 Report structure

The remainder of this report follows the following structure: first, the available data is collected and explored to understand the value and restrictions of the available data. This collection and exploration is executed in chapter 2. Additionally, the data is preprocessed in order to fit the research goal. Chapter 3 selects the most effective methods via a literature review to deal with the preprocessed data. This chapter evaluates 15 different methods and select the 2 most effective methods. Evaluation measures applicable to the selected anomaly detection methods are collected in chapter 4. Afterwards, chapter 5 discusses the general concept which will be executed to identify the anomalies. The evaluation measures will be connected to the anomaly detection method, to indicate the best design of such a method. This general concept is translated to the situation of Nexperia in chapter 6. An anomaly detection model is designed, including design choices and hyperparameter optimization. This model will be executed on the preprocessed data collected in chapter 2 and results will be presented. To elaborate more on the feasibility of the detection model in the organization, a feasibility and validation study is executed. This study identifies whether the implemented method is executable on a large scale and whether the method can be generalized for other customers. These results will be discussed, and implications will be considered. Finally, chapter 7 will draw final conclusions from the discussion in chapter 6 and based on these conclusions, recommendations will be presented to Nexperia.
2 Data Understanding and Preparation

To deal with business problems concerned with large amounts of data, the first step of developing a solution is to understand the data and connect the research goal with the data [39]. This chapter will evaluate the current format and properties of the data exchange at Nexperia. Subsequently, the value of the data regarding the research goal will be evaluated. Eventually, we will introduce an aggregation of the EDI to simplify the detection process, while maintaining the highest accuracy possible.

2.1 Customer forecast data exchange

As discussed, customer demand forecasts are exchanged between the Nexperia and their customers, to get insight about the future demand for Nexperia. To deal with the research goal, first the data should be collected and explored. The customer forecast data is received from downstream customers on an irregular basis. For simplicity purposes we assumed that all data is received weekly and that the detection of the anomalies is executed on a weekly basis as well. A simplified visual representation of an EDI received in a time series format can be seen in Figure 2.1

![Figure 2.1: An imaginary EDI received by Nexperia](image)

Figure 2.1 shows an imaginary EDI made in Week X for the Y-Z product-customer combination. On the X-axis the variable ‘rolling week’ is represented. The rolling week represents the week in which the forecast is received. The forecast that is received in week X contains future demand forecasts for the coming n ‘planning weeks’. The value of n differs between customers, with a maximum of 52 weeks. This value of X shifts with 1 each week. The Y-axis represents the forecasted demand for the corresponding weeks. EDI’s are unique for each product-customer combination. Nexperia expects to have around 15000 EDI’s entering the organization on a weekly basis for different product-customer combinations.

As stated in chapter 1, these customer demand forecasts are extracted from downstream customers on a weekly basis. This means that every week a similar time series (as shown in Figure 2.1) is extracted with a rolling week concept. The relation between each product-customer combination and the EDI is a
1-many combination, where each combination contains EDI’s for multiple consecutive rolling weeks.

As can be seen in Figure 2.1, it is difficult to classify future forecasts as anomalous or not. Firstly, employees do not have extensive decision rules to exclude/include observations. Currently, customer demand forecasts are only classified as anomalous if the total demand for first 13 planning weeks changes with 33% between consecutive rolling weeks. Additionally, after the initial data exploration, lots of large changes arise in the explored data. An example of large changes in consecutive weeks is visualized in appendix A. This appendix shows three consecutive rolling weeks for the next 52 planning weeks. These figures show that the current anomaly detection method based on the first 13 weeks principle would not have identified this potential anomaly. Additionally, it shows that large forecasts can fluctuate between consecutive rolling weeks.

High variability is common in these EDI’s, which shows that anomalies are hard to classify correctly. An anomaly detection approach should be identified that can deal with frequent fluctuations. If a simple changepoint detection would be used, these fluctuations would result in many anomalies. This issue shows that the value of the data in the current format is difficult to extract. After discussing this implication with demand managers of Nexperia, an aggregation concept is developed, based on their current classification process. This concept is called the cumulative time series, and is constructed from multiple EDI’s. This construction is discussed in the next section.

### 2.2 Construction of cumulative time series

Currently Nexperia collects the EDI forecasts from their customers on a regular basis. As discussed in the previous chapter detecting anomalies in the EDI’s is difficult and ineffective. Therefore, to make detecting anomalies more approachable, we introduce the cumulative time series concept. This concept is called the Cumulative Demand Forecast. Each product-customer combination has a 1-1 relation with the cumulative demand forecast. This is in contrast with the 1-many relationship of the EDI’s. Figure 2.2, shows a visualization of an imaginary cumulative demand Forecast and the construction process for 2 consecutive observations is explained below.

The cumulative time series is constructed from multiple historical EDI’s over time, collected for the same product-customer combination. Each observation on the cumulative time series, is constructed from one EDI. All forecasted demand from the EDI is summed and the total demand of one EDI represents one observation in the cumulative time series. Cumulative Demand Forecasts have a 1-many relation with EDI’s, while EDI’s have a 1-1 relationship with a cumulative demand forecast.
Visualizing the cumulative time series can be extremely helpful, because large errors can be located immediately by looking at the time series. However, as explained in the organizational description, Nexperia wants a model that can automatically detect unexpected observations.

The main goal of this cumulative time series is to conceptualize the large volume of EDI’s with high variability into a single concept with more stable demand. Relevant statistical properties are easier to compute from the Cumulative Demand Forecast as well. The advantages of the Cumulative time series in comparison to EDI’s are:

- Large changes in total demand between weekly forecasts are easier to identify.
- It enables the utilization of larger amounts of historical information, while keeping the computation time relatively low.
- It neglects the large variability currently present in the EDI’s.
- The currently used anomaly detection rules are based on the cumulative demand as well.
- It is easier to identify trends in the cumulative time series.

### 2.2.1 Concept evaluation

To ensure that the introduced concept is valuable for the organization, feedback from the organization is processed. The concept in general was received positively, because currently, the cumulative demand (for the first 13 weeks) is already used to detect large anomalies. Nexperia indicated that forecasting
intervals are important when identifying the value of an anomaly detection method. Forecast for the coming weeks are more important for planning employees directly related to the operational planning. Forecasts for the distant future can be used for a more tactical planning in order to visualize future demand and make purchasing decisions. Therefore, the organization proposed to evaluate different forecasting intervals individually. This evaluation will be used in chapter 6 to make a distinction between short term forecasts and long term forecasts.

2.3 Data preprocessing

The beforementioned sections discussed the general behavior of the customer forecast data, and introduced the aggregated cumulative demand forecast concept. This section discusses the inclusion/exclusion of data characteristics and the conceptual construction of the cumulative time series. Additionally, an identifier is introduced to connect the correct EDI’s to the cumulative demand forecast. Finally, outdated, and irrelevant data is excluded to ensure a smooth execution with relevant data.

2.3.1 Data characteristics

The collected data contains 20 different column variables each for one observation in the EDI. Which means that one EDI with forecasts for the coming 52 weeks is saved in 52 rows. Appendix B shows the selection of column variables and their reason for inclusion. The final result of the selection process is visualized by table 2.1.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID-code</td>
<td>Id-code constructed from the CPN &amp; Ship_to</td>
<td>CPN_Shipto</td>
</tr>
<tr>
<td>Rolling Week</td>
<td>The week in which the forecast is received</td>
<td>YYYYWW</td>
</tr>
<tr>
<td>Planning Week</td>
<td>The target week in which the demand should be fulfilled</td>
<td>YYYYWW</td>
</tr>
<tr>
<td>Forecast QTY by Nexperia</td>
<td>The forecast after evaluation</td>
<td>Integer</td>
</tr>
</tbody>
</table>

First, to ensure that correct data can be combined, ID-codes are introduced. These ID-codes are constructed from the Customer Part Number (CPN) and are combined with the ShipTo number. This results in a combination ID-code of xxxxxxxxxxxxx_yyyyyy (12 by 6 combined with an underscore). This ensures that the correct weekly forecasts are linked to the relevant cumulative time series. Additionally, this ensures that the forecasted demand is only linked to one product-customer combination. This ID-code serves as an identifier, because the relation between the EDI’s and cumulative demand forecast could be valuable after the detection process is executed.

The ’Rolling Week’ and ’Planning Week’ attributes respectively indicate in which week the forecast has been received, and for which week the forecast is intended. The Forecast quantity by Nexperia, shows the historical forecasts that have entered the planning engine.

An example of an EDI construction is as follows:

1. First select all data with the same ID-code
2. Select the Rolling Week of which the EDI is desired
3. The Y axis shows the Forecast QTY by Nexperia for each planning week on the X-axis.
An example of an EDI from the collected dataset is shown in Figure 2.3

![EDI forecast](image)

**Figure 2.3: A visualization an EDI forecast**

A parametric representation of the EDI is constructed below:

- Rolling week = R noted as week number.
- t = amount of weeks between rolling and planning week.
- Forecasted demand for planning week R + t = \( x_{R+t} \)
- A total EDI time series can be represented by: \( x = \{ x_{R+t}, \ldots, x_{R+n} \} \)

### 2.3.2 Cumulative construction

As discussed before, the translation from the individual to the cumulative time series has been made because in the individual time series, it is more complex to define an anomaly. This complexity resides in determining a relevant change between customer demand forecasts in concurrent weeks. Therefore, the following variables are used to transform multiple EDI’s to one Cumulative Demand Forecast (CDF).

- The CDF time series can be represented by \( X = \{ X_R, \ldots, X_N \} \)
- The cumulative forecast \( X_R = \sum_{t=1}^{n} x_{R+t} \)
  
  With rolling week R, and \( x_{R+t} \) extracted from the EDI time series explained above.

### 2.3.3 Data exclusion

Several complete product-customer combinations were excluded from the dataset. After the construction of the cumulative time series, 262 time series were present in the data. Two exclusion rules were implemented:

- Outdated time series (42 excluded)
  
  These time series were excluded because no historical data was available during the last month. Therefore, the product-customer combination was assumed to be outdated.
- Insufficient observations (8 excluded)

Several time series only contained 1-3 observations in the cumulative time series. It is impossible to effectively use the historical data because of insufficient information. Therefore, only cumulative time series with a minimum of 5 observations were included.

The final preprocessed dataset consists out of 212 time series from the initial 262 time series.

2.4 Summary

This chapter focused on the collection, understanding and preprocessing of the data. First the data was collected and explored. After the conclusion that the data in the current format had more complications than it provided value, an aggregation concept was introduced. This concept was evaluated and proved to be relevant. Eventually, the preprocessing and exclusion activities were discussed. These activities resulted in a relevant set of 212 time series which will be used in the remainder of this thesis. These 212 time series were finally reviewed with a planning expert to determine anomalies in this data.
3 Method Selection

This chapter will solely focus on research question 1: ‘What existing anomaly detection methods exist that are suitable for time series data’. To identify existing anomaly detection methods applicable to time series, a literature review has been conducted and the review will be discussed in section 3.1. This literature review resulted in 15 research papers that introduced relevant anomaly detection methods all applicable to time series data. Subsequently, from these 15 research papers, the most effective methods for the problem situation at Nexperia are selected in section 3.2.

3.1 Anomaly Detection methods

This section is used to introduce the gathered research papers. The relevant research papers each introduce at least one relevant anomaly detection method. However, several methods use combinations of detection concepts, hence, the papers are used to extract the main anomaly detection concepts that are used in the research papers.

The extracted research papers are collected in Table 3.1. Regarding the objective of an anomaly detection method, an important division can be made; detecting individual observations or detecting ranges of observations. After this division, two main categories of anomaly detection methods exist: ‘Machine Learning Methods’ and ‘Forecasting Methods’. Both the objective as the main detection category are included in Table 3.1 as well. In the next subsections, the categories are explained with academic examples of the categories from the research papers. Additionally, the strength and weaknesses of the extracted anomaly detection methods are discussed. Besides these categories, several alternative methods exist, of which some insights are discussed.

<table>
<thead>
<tr>
<th>Number</th>
<th>Detection method</th>
<th>Objective</th>
<th>Detection category</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CUSUM-EWMA statistical method</td>
<td>Ranges</td>
<td>Else</td>
<td>[11]</td>
</tr>
<tr>
<td>2</td>
<td>Piecewise Median Anomaly Detection</td>
<td>Ranges</td>
<td>Else</td>
<td>[44]</td>
</tr>
<tr>
<td>3</td>
<td>moving window k-Std dev</td>
<td>Ranges</td>
<td>Else</td>
<td>[8]</td>
</tr>
<tr>
<td>4</td>
<td>SCHEDA: Lightweight Euclidean dist</td>
<td>Both</td>
<td>Else</td>
<td>[21]</td>
</tr>
<tr>
<td>5</td>
<td>Seasonal ARMA-based SPC charts</td>
<td>Ind obs</td>
<td>Forecasting</td>
<td>[27]</td>
</tr>
<tr>
<td>6</td>
<td>EGADS by Yahoo</td>
<td>Both</td>
<td>Multiple</td>
<td>[30]</td>
</tr>
<tr>
<td>7</td>
<td>LSTM</td>
<td>Ind obs</td>
<td>Machine learning</td>
<td>[43]</td>
</tr>
<tr>
<td>8</td>
<td>HTM/LSTM models, bayes mixtures</td>
<td>Ind obs</td>
<td>Machine learning</td>
<td>[14]</td>
</tr>
<tr>
<td>9</td>
<td>Neural net ARIMA</td>
<td>Both</td>
<td>Machine learning</td>
<td>[10]</td>
</tr>
<tr>
<td>10</td>
<td>Hierarchical Tempory memory (HTM)</td>
<td>Both</td>
<td>Machine learning</td>
<td>[1]</td>
</tr>
<tr>
<td>11</td>
<td>DeepAnt anomaly detection</td>
<td>Both</td>
<td>Machine learning</td>
<td>[36]</td>
</tr>
<tr>
<td>12</td>
<td>ARIMA with ADAM</td>
<td>Ind obs</td>
<td>Forecasting</td>
<td>[32]</td>
</tr>
<tr>
<td>13</td>
<td>Regression and Bayes classifier</td>
<td>Ind obs</td>
<td>Forecasting</td>
<td>[3]</td>
</tr>
<tr>
<td>14</td>
<td>Predictive ARIMA</td>
<td>Ind obs</td>
<td>Forecasting</td>
<td>[42]</td>
</tr>
<tr>
<td>15</td>
<td>Comparison of 6 different statistical methods</td>
<td>Multiple</td>
<td>Multiple</td>
<td>[38]</td>
</tr>
</tbody>
</table>

As discussed before, anomaly detection methods can have two main objectives: detecting individual observations as anomalies, or detecting ranges of observations as anomalous. This objective is noted in column ‘Objective’, where ‘Ind obs’ indicates that the method can be used to classify individual observations. ‘Ranges’ is used to indicate the classification of time series intervals. Finally, ‘Both’ indicates that the method can be applied to both, time series ranges as individual observations. The column ‘detection category’ indicates if the method relates to one of the main detection categories: ‘forecasting’, ‘machine
learning’, or ‘else’. The final research paper [38] introduces six different detection methods, however, the detection results are discussed on a high level, and the individual methods are not discussed in great detail. Therefore, this paper can only be used to compare the results with the other research papers.

The following subsections discuss the two main method categories: Machine Learning methods and forecasting based classification.

### 3.1.1 Machine Learning methods

The first category of anomaly detection methods is the Machine Learning (ML). These methods can differ based on the method used, but all use a similar general concept. This concept uses information (training data) to train a detection model, which subsequently is used to classify the anomalies. For anomaly detection in time series with ML models, labels are extremely important. These labels contain the information about whether a time series or observation is an anomaly. These labeled time series are used as training data, in order to train a model what anomalous behavior is.

The advantages of ML models, are that the models often have a high predictive accuracy. Which means that compared to other types of anomaly detection models, ML models outperform the competition. Additionally, ML models are extremely robust because most ML models, iteratively optimize their hyperparameters. This ensures that the model adapts over time, when new behavior is present in an organization. This is also denoted by the ‘learning’ aspect of ML models. However, disadvantages of the ML models are that lots of labeled training data needs to be present. ML-models are trained based on training data, and it is important to include high quality training data. If low quality data is used as training data, then the predictive capability of the system is lower, because the model would 'learn' the inaccurate behavior [23]. Therefore, high quality training data is the most critical aspect of machine learning methods. Additionally, the execution time of ML models, can be extremely fast, while the training time is generally long.

One of the disadvantages of ML models is the interpretability. ML models use abstract concepts that often are unexplainable. A common concept is the use of neurons in neural networks, which are variables that change in value each time new training data is presented to the model. These neurons can be trained to detect anomalous behavior in a time series, based on historical information. This alteration of variables, results in issues with interpretability. These issues can be explained by the black box and white box concept. White box models use interpretable formulas and computations to achieve an understandable final result. However, in black box models (which most ML-models are) the computations are impossible to trace. The concept that ML-models use is that the model changes these neurons extensively, which leads to the inability to interpret the computations. As discussed before, lots of training data is needed for an ML model to perform well, which also would lead to millions of hyperparameter tweaks. It would be impossible to determine what types of training data, would lead to what kind of tweaks in the hyperparameters.

To conclude, machine learning methods are high performing anomaly detection methods, that are able to change based on new data. However, to utilize an ML model two critical prerequisites are necessary such as high quality training data, and the organization should accept the utilization of the black box concept.

### 3.1.2 Forecasting based classification

Forecasting methods are often used in demand planning or supply chain planning to forecast behavior in the future. However, in anomaly detection, forecasting is often used to model the past. In anomaly detection, forecasting methods are used to model time series behavior based on historical data [3] & [42]. The goal of this modeling is to specify normal behavior. This normal behavior can subsequently be used to classify new data as anomalous or not, based on the historical characteristics.
Two main forecasting categories exist to model the normal behavior, with lots of variants based on the main categories. The main categories are discussed with their advantages and disadvantages. Subsequently, the classification activity based on this normal behavior is discussed. Additionally, several modeling choices are introduced which can optimize the performance of such models.

**Exponential smoothing**

The first category is exponential smoothing methods, where an exponential function is fitted through the historical data based on the moving average principle [24]. This exponential function at moment $x$ is based on an exponential window, where recent observations have a higher weight than more outdated observations. This concept is applied to historical data in a time series to fit a smoothed function. This smoothed function is assumed to be normal behavior, and based on the characteristics of this normal behavior, data can be classified as anomalous or not. Several different alternatives based on exponential smoothing can be utilized, where trend, seasonality and variability can be modeled in more detail to optimize the performance.

**Regressive methods**

Regressive methods are the second category, where similar concepts are applied as in the exponential smoothing category, however the fitted model is constructed in a different way. Regressive methods apply regression based fitting based on historical windows to forecast the new observation. This enables the method to more precisely fit a function from the historical data. These methods often are more accurate but have a larger computational footprint than exponential smoothing methods.

**General forecasting insights**

In both beforementioned categories, lots of design choices arise. Hyperparameters, trend, seasonality, and variability are aspects to take into account when selecting a method. Both methods are similar however, several differences exist. Regressive methods are often higher performing methods, but have a lower computational efficiency [2]. Especially, when the regression based time-window is large, computations can take a lot of time. Exponential smoothing methods are more computational efficient, but often have a lower performance, because when high variability exists in the data, this highly effects the fitted model. The reason for this is that the model solely uses the moving average concept.

To conclude, these differences between the models should be taken into account while selecting one of the models. If accuracy is the main focus, regressive methods should be selected. But if computation time is a main factor, exponential smoothing methods could be a relevant alternative. Both methods have their own approach in dealing with trend and seasonality, so these design choices should be taken into account when selecting the best variation for both categories.

**3.1.3 Forecasting classification process**

After the modeling of normal behavior with forecasting methods, this behavior can be used to classify new observations as anomalous or not. The difference between the forecasted values and the actual observations is used as information about the history. The differences between the forecasted values and actual observations are called residuals. Residuals can subsequently be used to evaluate variability, trend and seasonality in time series. These characteristics can be used to compute classification boundaries for new observations. These classification boundaries are often based on statistical properties of the residuals. Two different approaches to determine the classification boundaries exist: parametric classification and non-parametric classification.

**Parametric classification**

Parametric classification is the most effective method when the time series have similar distributions (normal/exponential/gamma). When classification boundaries are selected, it is necessary that they are generalizable for multiple time series. Especially, because Nexperia aims to evaluate 15000 different time series on a weekly basis. Therefore, the similarity in distribution between residuals is important.
Otherwise classification boundaries could be very inaccurate. If the residuals cannot be explained by similar distributions, the residuals could be transformed with log, normalized, or min-max methods. If the residuals still cannot be explained by similar distributions after the transformations, non-parametric classification should be used.

**Non-parametric classification**

Non-parametric classification uses different methods to classify observations. Where parametric methods are often focused on the mean and its distribution from the mean, non-parametric classifications use the median to indicate the ‘reference point’. The median can account for large outliers in the residuals, because these outliers do not have a large effect on the distribution with non-parametric classification. Two non-parametric classification methods are: boxplots or distance based classification.

### 3.2 Method selection

As can be concluded from the literature review conducted in section 3.1, the objective and the restrictions of the anomaly detection method should be taken into account. The objective of the anomaly detection method is clear; 'Classify individual time series observations as anomalous or not'. The remainder of this section discusses the organizational restrictions, whereafter the most relevant techniques are selected.

#### 3.2.1 Organizational restrictions

Chapter 1 discussed the current situation at Nexperia. This included the availability and presence of historical data and the desired outcomes of the anomaly detection model.

1. An anomaly detection model that automatically can classify anomalies from a time series format.
2. The method should be able to classify single new observations, instead of ranges of observations as anomalies based on their historical behavior.
3. The method should be able to predict anomalies without prior knowledge about the identity of anomalies (anomaly labels).
4. The desired outcome is an anomaly detection method that should be understandable by employees with a large background diversity.
5. To ensure robustness, the model should be able to adapt itself (Machine learning /automatic parameter optimization).
6. Each combination has their own behavior and distribution. Several distributions include a trend, and the model should be able to reach a high accuracy for all time series variations.

#### 3.2.2 Selection procedure

To select a method (or part of a method) from the literature review, the beforementioned restrictions are taken into account. Selecting the best method is a difficult process, because the used authors have executed their anomaly detection methods on different datasets. Additionally, the authors model the methods specifically, for their problem situation. Therefore, the main focus is to adhere to as many of the requirements as possible. To select the most effective method(s) a structured approach is used by excluding methods that cannot fulfill the most important organizational requirements. Subsequently, Table 3.2 shows the structural selection process in which methods are excluded. The exclusion reasons based on the abovementioned requirements are visualized in the 'Exclusion reason' column, whereafter the reasons are discussed in more detail.
Table 3.2 shows the evaluated methods, and their exclusion reasons. The main exclusion reasons are discussed.

**Anomaly labels and understandability**
Nexperia does not know about their historical observations whether or not the observations were correct or false. Therefore, an anomaly detection method should be used that can classify anomalies without prior knowledge of anomalies. Most machine learning methods need an enormous amount of training data that includes knowledge about anomalies. As [23] states, a ML model is as good as its training data. In addition to the presence of anomaly labels. Methods 7, 8, 9, 10, and 11 are more difficult to understand because of interpretability difficulties. These methods are based on machine learning concepts and introduce many complications. As stated before, Nexperia requires a model that is understandable by multiple background diversities. However, machine learning methods are abstract and hard to understand, which can implicate relevant improvements in the future. Furthermore, machine learning methods can be seen as black boxes, the computations that were used to classify anomalies cannot be traced back to the cause, which is a limitation for Nexperia. Therefore, the methods, 7, 8, 9, 10, and 11 are excluded from the detection method subset.

**range detection**
Several research papers use ranges of observations to detect anomalies. However, differences arise in these ranges. Methods 1, 2, and 3 are only able to classify ranges of observations as anomalous. This ensures that often these methods are slow in their detection process. Because the statistical properties can take multiple observations to reach the anomaly threshold. Although method 4 uses observation ranges as well, this method uses the ranges to learn historical behavior in order to classify both ranges or individual new observations. Therefore, this method was not excluded from the subset. But methods 1, 2, and 3 were excluded.

**Final detection method**
The resulting research papers are method 4, 5, 6, 12, 13, and 14. The authors of paper 6 introduced a java package with a broad selection of effective anomaly detection methods. However, because of the desire to implement the anomaly detection in an environment where inexperienced programmers exist, the most adoptable programming language is selected, which is Python [22]. Therefore, the java package of research paper 6 is excluded. Research paper 5, 12, 13, and 14 have utilized regression based...
forecasting to model their historic behavior. In addition they used a classification method to eventually classify new observations as anomalies or not. Because of the corresponding performance of the regressive methods experienced by these papers, we will use forecasting methods to model historic behavior. However, to prove that regressive forecasting methods are superior to other forecasting methods, an exponential smoothing variation will be evaluated as well to compare the performance of the regression based detection.

Finally, for the classification process parametric classification methods are recommended. Because regressive methods assume that the resulting residuals are normally distributed. However, the data understanding phase showed that the historical data presented incidental large fluctuations. These fluctuations could impact the presence of normally distributed residuals. To ensure generalizability while accounting for the presence of anomalies in the training data, the non-parametric classification method ‘boxplot’ will be used.

### 3.3 Conclusion

Different types of anomaly detection methods exist in literature. The main categories are Machine Learning, and forecasting based classification. ML methods have many advantages such as high computational efficiency and generally high performances. However, the numerous drawbacks of ML method such as necessity of large volume of training data, the necessity of labeled data, and interpretability issues, have resulted in the exclusion of ML methods.

Instead of an ML method, forecasting methods were assumed to be more feasible for our problem situation. Different forecasting methods exist, however, from literature the Autoregressive Integrated Moving Average (ARIMA) method was the best performing method when seasonality is absent in the data. However, ARIMA has several drawbacks, such as computational efficiency. Therefore, Double Exponential Smoothing (DES) is selected to compare the performance of the models, while introducing a faster forecasting method. This method achieves lower accuracy in literature but a higher computational efficiency. These forecasting methods will be supplemented with the non-parametric classification method ‘Boxplot’. This method will be used to eventually classify the observations.
4 Evaluation technique selection

To efficiently execute the selected anomaly detection method, several hyperparameters have to be selected. The selected anomaly detection method consists out of two phases and evaluation methods for both phases should be selected. First, the evaluation techniques for the forecasting performance will be discussed. These techniques will evaluate the accuracy of the selected forecasting techniques. Secondly, the classification evaluation will be discussed. The evaluation measures for the classification process will be the most important, because they will evaluate the performance of the final detection model.

4.1 Forecast evaluation measures

The first step of the anomaly detection method selected in chapter 3 is to model normal behavior based on historical information. As briefly discussed before, this is executed by a forecasting method. Two forecasting methods are selected and these are both evaluated with the same evaluation techniques. Several alternatives are discussed below, in order to give a comprehensive selection of forecasting evaluation techniques.

4.1.1 AIC/BIC

The most often used measures to evaluate how well the original behavior is modeled by a forecasting method are the Akaike information criterion (AIC) and Bayesian information criterion (BIC). Both models can efficiently evaluate the accuracy of a parametrized model. However, the criteria perform poorly when comparing time series with their differenced versions [4]. This method would be suitable for Double Exponential Smoothing, however, ARIMA introduces differencing in order to ensure the degree of stationary (which is an assumption of ARIMA). Therefore, both the AIC and BIC cannot be used in this thesis.

4.1.2 Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE), is a simple measure that utilizes the difference between a forecasted value and the actual observation. The measure is computed by the following equation:

\[
\text{MAPE} = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right|
\]

\(A_t\) represents the actual observation, \(F_t\) is the forecasted value of the model, and the \(n\) is the number of observations. It can be directly seen that this measure results in challenges when applied in reality. The \(A_t\) variable is used as a denominator, which in several domains such as demand forecasts, can often be 0. This results in a equation that is unsolvable. This measure is simplistic which is nice to interpret, however, in domains such as demand forecasting (which is the focus of this thesis) this measure is not effective. To account for this issue, several adjusted measures are introduced in literature [29]. However, additional drawbacks of the MAPE is that it often produces a skewed measure for high errors during low demand periods. Many research papers address this measure or any adjustments on it as easy to interpret but poor performing measures. This results in the exclusion of these measures for this thesis.

4.1.3 Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) is a forecasting measure that uses the following equation to compute the measure:

\[
\text{RMSE\%} = \sqrt{\frac{\sum_{t=1}^{n} (A_t - F_t)^2}{n}}
\]
This measure uses the mean squared error which weights large errors massively because of the squared difference. However, because our training data contains potential anomalies, this measure could exclude decent forecasts with a single large outlier. In such a case the average error would be skewed largely towards the single outlier. This is not desirable because this would not represent the intended historical behavior well. Therefore, this measure is excluded as well.

### 4.1.4 Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is a simple measure as well, which computes the distances between the forecasted value and the actual observation. Which is represented by the following equation:

\[
\text{MAE} = \frac{\sum_{t=1}^{n} |A_t - F_t|}{n}
\]

This measure accounts for the issues with the MAPE, because it is able to utilize demand forecasts with a demand of 0. However, this measure has drawbacks as well. For example, when evaluating a MAE it is not directly clear if it is a good performing forecast or not. A MAE of 10 can be good performing if average demand is 5000, however when the average demand is 30, a MAE of 10 is poor performing. However, this drawback does not apply at our problem situation, because this measure is used to compare time series with the same explanatory time series. This ensures that the comparison is always relevant. Therefore, this measure is used to evaluate the forecasting performance.

### 4.2 Classification Evaluation

The final step of the anomaly detection method is the classification of anomalies. This is executed by the non-parametric classification method 'boxplot'. This method is used to classify all time series. The classification is evaluated in order to assess the detection performance. A confusion matrix can be used as general concept to indicate which anomalies are correctly detected. Based on this confusion matrix, several evaluation measures are used to indicate the effective performance of the anomaly detection.

#### 4.2.1 Confusion matrix measures

A confusion matrix is a general concept which is consistently used in anomaly detection evaluation throughout the extracted research papers [6], [9] & [3]. A confusion matrix identifies the predicted values of an anomaly detection method and compares these predicted values against the real values and expresses these values into rates/values. The resulting rates and values are the True Positives (TP), the False Positives (FP), the False Negatives (FN), and the True Negatives (TN). To utilize these evaluation measures, anomaly labels should be present. Otherwise it would be impossible to classify something as correctly /incorrectly predicted. From these values multiple evaluation measures are constructed like the 'Accuracy, Recall, Precision', and F₁ score, which is discussed below:

**Accuracy**

The accuracy is used by the following research papers: [6], [9], [16], [19], [25], [30], [32], [38], [43], [45]. The research papers introduce the accuracy as a simple measure, which illustrates the amount of values correctly predicted. It does not take into account the rates in which the method predicted anomalies incorrectly, or neglected anomalies at all (False Positives and False Negatives).

The research papers utilized the accuracy with different approaches. Several research papers stated that accuracy values higher than 0.80 are assumed sufficient [6], [32]. Others only included the fact that the accuracy was used, but did not include numeric results [16], [38]. The largest share of research papers use the accuracy measure solely to compare anomaly detection model performances [9], [19], [25], [30], [43], [45]. The accuracy is only valuable to compare models, not to indicate performance in general, because it neglects the beforementioned False Positives and False Negatives.
F-score
The precision (accuracy rate) and recall (recall rate) are used to compute the F-score which is often regarded as the most effective single performance measure as stated by [14], [36]. It utilizes all confusion matrix measures: the True Positive, True Negative, False Positive, and False Negative rates. This ensures that the F-score includes all information about the performance of an anomaly detection model.

The F-score is used by a large share of research papers [6], [9], [11], [14], [19], [25], [30], [34], [36], [43], [44], [47]. These research papers adjusted the F-score to their situation. The function for the F-score is:

$$F_{\beta} = \left(1 + \beta^2\right) \cdot \frac{\text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}$$

The $\beta$ variable indicates which variable is more important, the precision or recall. This can be utilized to put more importance towards recall ($F_2$), where $F_{0.5}$ increases the importance of the precision. In practice, these different values for $\beta$ are used to regulate the importance of False Negatives values.

The F-score is a measure which is not useful on its own because it is a comparison between two other measures (precision, recall). But it is highly effective when comparing the performance of different methods. This is utilized by most research papers, where the F-score is often the base for visualizations like the ROC curve.

ROC and AUC
The final measure that is analysed is the Receiver Operating Characteristic (ROC) in combination with the Area Under the Curve (AUC). ROC curves are curves based on the confusion matrix measures where the Y-axis represents the True Positive rate and the X-axis represents the False Positive rate. The AUC is a single numeric value which increases the availability to compare the ROC curves between anomaly detection methods. This measure is constructed from the area under the ROC curve [17]. The following research papers use the ROC and/or AUC: [6], [21], [25], [35], [36], [45], [46], [47]. This graph is commonly used to represent the performance in regard to random guessing in prediction/detection models.

4.3 Conclusion
Section 4.1 & 4.2 discussed the importance of the evaluation measures. After weighing the advantages and disadvantages of the evaluation measures, the Mean Absolute Error is selected. AIC and BIC are excluded based on the inability to deal with differenced time series, that have different number of parameters. Subsequently, the Mean Absolute Percentage Error is excluded because of the inability to deal with zeros in the forecasts. Finally, a selection is made between the RMSE percentage and the MAE percentage. After the initial data exploration and preparation, several large potential errors were located in the historical data. The RMSE locates a high weight to the existing errors because of the squared characteristic. However, this would not be beneficial for the overall model, because errors in the data should not heavily impact the fit of the model. Therefore, after excluding all other alternatives, the Mean Absolute Error is selected as evaluation measure for the forecasting process. Both the regressive and smoothing performance are evaluated with this evaluation measure in order to effectively compare the forecasting results.

After discussing the used classification evaluation measures, it is necessary to assess which measures are useful for our problem situation. The F-score and ROC curve are the most effective single evaluation measures. However, they require the total knowledge of the confusion matrix. This means that all rates (TP, FP, FN, TN) should be known. To make this research as comprehensive as possible, these measures will be used for a small subset of time series. Of this subset the presence of anomalies will be manually evaluated. Therefore, the F-1 score will be used to give an indication about the classification boundaries.

These evaluation measures are effective in evaluating the forecasting and classification performance. However,
Laptev et al. [30] introduce a trade-off which should be taken into account. Anomaly detection methods can be designed to achieve the highest possible accuracy. However, computation time and memory usage in the databases can be extremely large in achieving the highest accuracy. Therefore the authors state that a trade-off between accuracy, memory usage and training time should be taken into account when designing an anomaly detection model. This should ensure that the anomaly detection model is practically applicable.
5 Anomaly detection execution

After preprocessing the data, selecting the anomaly detection method, and identifying evaluation measures, this chapter will combine the three chapters to introduce the anomaly detection concept. First an overview of the detection method will be introduced in section 5.1. Whereafter the individual steps are explained in more detail. Section 5.2 will discuss the modeling of normal behavior by the forecasting methods. Subsequently, the classification concept will be discussed in section 5.3 to explain the classification of observations. This chapter will not discuss design choices, and the situation at Nexperia will not be taken into account. The objective of this chapter is to purely discuss the individual steps of the detection method. The next chapter will discuss the implementation of this method for the situation at Nexperia.

5.1 Execution overview

The selected anomaly detection method can be divided into three main steps:

- **Modeling normal behavior**

  First, historic observations from the cumulative time series are used to model normal behavior. This behavior is computed by the forecasting methods selected in chapter 3. For each observation on the cumulative time series, an value is computed based on the observations prior to that observation. This results in a new ‘computed time series’ which we note as normal behavior. The goal of this computation is to capture statistical characteristics of historical behavior.

![Figure 5.1: A simplified modeling of computed normal behavior for this time series](image-url)
• **Computing residuals**
  Secondly, deviations from the modeled behavior are computed by taking the difference between all observations and their respective computed counterpart. These differences are called the residuals and represent the distance between the computed normal behavior and the actual observations. It can be assumed that residuals are low for time series with a low variability. Because of the higher forecasting accuracy. Figure 5.2 shows a visualization of residuals. The residuals are always computed by subtracting the computed value from the original observation. This results in the possibility that residuals can both be positive as negative.

![Figure 5.2: Residuals computed from Figure 5.1](image)

\[
\text{Residual value} = X - F(X)
\]

• **Classifying observations**
  Finally, the residuals from the last step are used to compute statistical characteristics of historical behavior. These statistical characteristics are used to compute classification boundaries. This results in large classification boundaries for time series with high variability, and narrow classification boundaries for time series with low variability. Example classification boundaries are visualized in Figure 5.3. These classification boundaries are computed based on the boxplot principle. In the boxplot, median based characteristics for all residuals are computed. From these characteristics, classification boundaries are computed for the ‘to be classified’ observation. The classification boundaries are computed based on the interquartile ranges (IQR), which will be discussed in more detail in section 5.3.
5.2 Modeling normal behavior

As derived from chapter 3, forecasting techniques are effective to model normal behavior. The most effective techniques resulting from the papers are regressive methods. Because of comparison purposes we also evaluate an exponential smoothing method in order to prove that the regressive method is the most effective. As discussed before, these methods have different alterations for different time series. This section explains the concept of modeling normal behavior by both forecasting methods.

The introduced cumulative time series concept introduced in chapter 2, is used to model the normal behavior. Historical data is used in this process. The purpose of this normal behavior is to capture its statistical characteristics. The formula’s that are used for the computation of the ARIMA method are explained in appendix C. The DES method is a simple moving average with a trend component and will not be extensively discussed.

The forecasting methods have differences in taking account the trend and variability of the historical data. After the initial data exploration process in chapter 2, we have seen that seasonality is absent. Therefore, the selected forecasting methods will not focus on explaining seasonality. The following subsections introduce both forecasting methods.

5.2.1 Autoregressive Integrated Moving Average (ARIMA)

The regressive method that is used to model normal behavior is ARIMA. Multiple effective regression methods exist however, the problem situation should be regarded closely when selecting the method. The necessity to capture a trend in the data leads to the exclusion of a normal regressive method. Because trends are actively present and seasonality is lacking in the data, we will use the ARIMA method.
This method can deal with trends in the data but lacks the seasonal aspect. First the initial assumptions/preconditions of the ARIMA method are discussed.

**ARIMA assumptions**

The ARIMA method to model ‘normal’ behavior, has several assumptions which should be taken into account when computing the (p,d,q) parameters.

- **Stationary**
  The characteristics of the time series should not be different over time, when computing observations. This means that when the mean and standard deviation are computed at different moments in time, they should not significantly deviate.

- **the data should be univariate**
  ARIMA computes variables based on prior observations for one variable, therefore, multivariate data cannot be utilized.

The ARIMA assumptions should be taken into account when modeling the normal behavior. the ARIMA method has three parameters, the *Autoregressive*, *Integrated*, and *Moving Average*, noted as (p, d, q) respectively. The basic concept of ARIMA is a regression on it’s prior observations, which decreases in relevance over time. This relevance can be modeled by the autoregressive parameter. This parameter can specify the amount of prior observations that is taken into account to compute an observation. The integrated parameter can be used to account for trends in the data. The value of this observation shows the times the time series is differenced. Finally, the moving average parameter can be used to take into account the error observed in prior computations. This research will use the following structure to model normal behavior with the ARIMA method.

1. Account for the ARIMA assumptions by evaluating the degree of stationary in the time series by executing the Dickey Fuller test [13].

2. Select the lowest d for which the Dickey fuller test is significant, with a minimum of d = 1.

3. Select the additional hyperparameter configuration with the lowest MAE%.

These steps together are used to compute the hyperparameters of the ARIMA method. After the computation of the hyperparameters of the ARIMA method the computation should be executed. Each observation in the original time series receives a computed observation. Appendix C, can be consulted for the mathematical explanation of the ARIMA method.

**5.2.2 Double exponential smoothing**

The exponential smoothing method that is used is the Double Exponential Smoothing. This method is selected with similar arguments as the regressive method. This method can take into account trends, but lacks the ability to account for seasonality. This seasonality is neglected because in the initial data exploration no seasonal data seemed to exist.

The advantages of Exponential smoothing methods are that they are faster then regressive methods. Additionally, the only assumption that is used is that the data should be univariate. It can be concluded that this provides a larger flexibility because the stationary assumption of the ARIMA method can be neglected. This would ensure that non-stationary time series would be available to include in this research. However, because we compare both methods, only stationary time series are taken into account. In general these advantages weigh against the lower accuracy of exponential smoothing in comparison with regressive methods. These aspects are evaluated in the next chapter.
5.3 Anomaly classification

This thesis uses a non-parametric classification method, because after the initial data exploration, many potential anomalies already existed in the data. Therefore, as noted by [12], it is ineffective to use a parametric classification method. The main result of these errors in the historical data is that no similar distribution can be used to model all time series. Therefore, to keep the academic value of this thesis, the non-parametric boxplot if selected to classify anomalies.

5.3.1 Boxplot

A boxplot can be used as a non-parametric classification of anomalies. Boxplots are constructed from the median and corresponding quartiles. Boxplots can be used to classify data as an anomaly or not by comparing the residuals in a time series. When a residual is significantly divergent from the rest of the residuals, this residual is potentially an anomaly. The following characteristics of box plots are used in classifying anomalies:

- The median is the middle number out of the residual values.
- 1st quartile (median of the first half of the dataset)
- 3rd quartile (median of the second half of the dataset)
- Classification boundaries: The classification boundaries are variable, and can be chosen to serve a chosen purpose. This purpose can differ, but often used classification boundaries for anomalies are 1.5 * InterQuartile Ranges (IQR) for small anomalies, and 3 * IQR for large anomalies. A visualization of a box plot can be seen in Figure 5.4

As discussed the boxplot is used to capture statistical properties based on the median. Classification boundaries can be selected to eventually classify observations as anomaly or not. Multiple classification boundaries are used to achieve different goals. In anomaly detection, the 1.5 * IQR and 3 * IQR are often used to classify small and large errors respectively. However, 1% and 99% quantiles are occasionally used as well. This is a design choice in the design study and it can clearly be seen that the classification boundaries have an extremely large impact on the actual classification.

Figure 5.4: Construction of a boxplot
5.4 Pseudo algorithm

This section shows the Pseudo-code for an executed anomaly detection iteration. This would be executed once per week.

The executed steps are discussed below:

1. \( \text{CDFlist} = \text{List of Cumulative Demand Forecast each for a different product-customer combination.} \)
2. \( \text{CTS}_n = \text{Cumulative Demand Forecast with the identifier n.} \)
3. The best hyperparameters would be computed by the grid search and stored in a database from which, the parameters are easily obtained to increase computation speed (implementation explained in chapter 6).
4. \( F(X_n) = \text{Model normal behavior by computing values for all observations in the cumulative time series.} \)
5. Compute the residuals for each observation by subtracting the computed values from the original observation.
6. Construct a boxplot of the residuals for a cumulative time series, and compute the distance between the residual of the target observation and the boxplot.
7. If the distance is larger than the classification boundaries Upper Control Limit (UCL), and Lower Control Limit (LCL), then classify the time-series as an anomaly.

Table 5.1 captures the explained method in a pseudo algorithm.

Table 5.1: Pseudo code anomaly detection

<table>
<thead>
<tr>
<th>Algorithm 5.1: Classification of anomalies in a list of cumulative time series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect list of cumulative time series - ( \text{CDFlist} = [\text{CTS}_1, \text{CTS}_2, \ldots, \text{CTS}_n] )</td>
</tr>
<tr>
<td>for each time series ( \text{CTS}_n ) in ( \text{CTSlist} ):</td>
</tr>
<tr>
<td>Select best hyperparameters for used forecasting method, from the database</td>
</tr>
<tr>
<td>Compute a value ( F(X_n) ) for observations ([X_1, X_2, \ldots, X_n])</td>
</tr>
<tr>
<td>Compute residuals list ( \text{RES} = [\text{RES}_1, \text{RES}_2, \ldots, \text{RES}_n] ) by: ([X_1, X_2, \ldots, X_n] - [F(X_1), F(X_2), \ldots, F(X_n)])</td>
</tr>
<tr>
<td>Compute the statistical characteristics of the residual list</td>
</tr>
<tr>
<td>Compute the distance ( d ) from ( \text{Quart}_1 ) or ( \text{Quart}_3 ) expressed in number of IQR’s</td>
</tr>
<tr>
<td>If ( d \geq \text{UCL} ) or ( d \leq \text{LCL} ):</td>
</tr>
<tr>
<td>Classify time series as potential anomaly</td>
</tr>
<tr>
<td>Else:</td>
</tr>
<tr>
<td>Classify time series as no anomaly</td>
</tr>
</tbody>
</table>

This pseudo code is the base of our implementation of the anomaly detection method at Nexperia.
6 Implementation and Results

This chapter will utilize the anomaly detection concept introduced in chapter 5 and discuss the implementation at Nexperia. The first section 6.1 will justify several design choices in order to provide insight about the decisions made. The forecasting performance will be discussed in section 6.2. Section 6.3 will provide a visual overview of the implemented method in the organization. This will include the data collection, employee feedback and managerial impact of an anomaly detection method. This implementation will be evaluated by the evaluation measures from chapter 4. Finally the results will be presented in order to give a comprehensive understanding of the anomaly detection methods where both forecasting methods are evaluated separately.

6.1 Design choices

In the previous sections of this chapter the anomaly detection method is introduced, and the individual functions are explained. This section discusses several design choices. In this section three parameters are evaluated: 'Normal behavior horizon', 'Relevance difference between forecasts for the near and far future?', and 'Which classification boundaries are selected?'.

6.1.1 Time-dependent relevance - Normal behavior horizon

The first evaluated variable focuses on the number of historical observations used for the modeling of normal behavior? This decision is a trade-off between the computational efficiency and performance. The computational efficiency would perform best if the least amount of historical data would be included. However, performance should be the main goal when deciding the amount of historical data to include. Figure 6.1 illustrates the issues that can arise when demand follows a trend over time. This figure shows that when forecasting the future, recent observations are generally more accurate than observations from 2 years ago.

![Figure 6.1: The decreasing relevance of historical observations, when trends are present](image-url)
This indicates that the relevance of observations decrease over time. Both forecasting methods already do take into account decreasing relevance over time. However, to optimize the computational efficiency, the least amount of historical data should be selected. ARIMA has an assumption that the data should be stationary and this assumption is the base for the horizon selection. The stationary assumption is evaluated for all preprocessed data. Afterwards the smallest interval for which > 95% of the time series can be stationary is selected to be the interval horizon. We evaluate 13, 26, 39, 52, 65, and 78 past Customer Demand Forecasts as interval horizons. These horizons are subjectively selected based on the quarterly principle. Currently, the training data consists out of 212 time series, which is a small subset. It can be concluded that when applying anomaly detection on a larger volume of data, that this design choice has to be re-evaluated.

For this selection we use the Dickey-fuller test to evaluate the degree of stationary. If the Dickey-fuller test is significant (<0.05) then we assume that the interval is stationary. In the data exploration of chapter 2, several non-stationary time series were observed with steep trends. To deal with these time series, we use the differencing concept, which is used by ARIMA as well, to increase the degree of stationary. Therefore, we test the Dickey-fuller test for no, 1 and 2 differencing degrees. This accounts for trends in the data.

A limitation of this method is that the stationary assumption could be researched in more detail. By evaluating Autocorrelation plots, or mean and standard deviation over time. However, the Dickey-fuller test already takes these aspects of time series into account to a certain degree. Additionally, this anomaly detection should be applicable to 15000 time series each week, which shows that a general approach is necessary. Because it is extremely time consuming to manually select parameters for all time series.

Table 6.1 shows the results of the Dickey-fuller tests executed on the time series. The stationary numbers indicate the amounts of time series that were significantly stationary. After evaluating the results the number of historical forecasts that is taken into account is 39 observations. Because the difference between 26 observations and 39 results in a considerable improvement. However, to keep the computation time as low as possible, intervals with more than 39 observations are neglected, because they don’t result in significant improvements.

Table 6.1: Stationary Dickey-Fuller test

<table>
<thead>
<tr>
<th>Interval</th>
<th>Stationary count</th>
<th>Non-Stationary count</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>384</td>
<td>40</td>
<td>0.906</td>
</tr>
<tr>
<td>26</td>
<td>378</td>
<td>46</td>
<td>0.892</td>
</tr>
<tr>
<td>39</td>
<td>409</td>
<td>15</td>
<td>0.965</td>
</tr>
<tr>
<td>52</td>
<td>410</td>
<td>14</td>
<td>0.967</td>
</tr>
<tr>
<td>65</td>
<td>410</td>
<td>14</td>
<td>0.967</td>
</tr>
<tr>
<td>78</td>
<td>410</td>
<td>14</td>
<td>0.967</td>
</tr>
</tbody>
</table>

### 6.1.2 Relevance difference between forecast in the near and far future.

The second concept that is evaluated is the relevance difference between intervals in the EDI forecasts. Nexperia uses EDI’s which are received from the customers on a weekly basis. These EDI’s are forecasts for the forthcoming weeks, where every customer has different forecasting intervals. Which means that customer A can forecast for the coming 26 weeks, and customer B can forecast for 52 weeks in the future.

By applying the aggregation concept discussed in chapter 2 information about the EDI forecasts, gets neglected. Therefore, this concept is introduced to still take into account the relevance of the planning weeks from the EDI’s. Because of planning nervousness, forecasts in the near future generally have a higher accuracy and more relevance than forecasts in the distant future. This is why differences have to
be made between time-dependent forecasts. Currently, Nexperia evaluates forecasts for the forthcoming 13 weeks with a total demand deviation, as explained in chapter 1. After discussing this decision with a planning employee of Nexperia, the employee indicated that the first 13 weeks are most important. This importance is connected to the lead time of the products, because large errors for the coming 13 weeks can directly impact the lateness of products delivered to customers.

The existing 13-week concept is the base for this design choice. The cumulative time series are divided into two separate sub-time series. The first sub-time series represents the total demand for the first 13 weeks, where the second sub-time series represents the total demand after the first 13 weeks. This results in a clear distinction in importance, where the first 13 weeks are more relevant for the planning employees that are responsible for production planning. However, the total demand after the first 13 weeks can be used in a more tactical planning, to discuss machine acquisition, or operational improvements for the future.

A visualization of the distinction between EDI forecasts before and after the 13 weeks is shown in Figure 6.2. One sub-time series is constructed by summing the forecasts in planning weeks 'X - X+13'. The second sub-time series is constructed by summing the forecasts in planning weeks 'X+13 - X+n'.

![Figure 6.2: The relevance of using different intervals to construct the cumulative time series](image-url)

Figure 6.3 shows a CDF of a single product-customer combination, which is afterwards dissected into two sub-time series of the same product-customer combination in Figure 6.4. The green sub-time series shows the CDF of the first 13 weeks, and the black sub-time series shows the CDF for the interval after week 13. This concept is applied to all product-customer combinations to ensure the relevance difference. The anomaly detection method explained in chapter 5, is executed for both sub-time series for each product-customer combination.
6.1.3 Classification boundaries

Subsection 6.1.1 & 6.1.2 both show different design choices focused on the forecasting performance that is included in the implementation for Nexperia. After the forecasting and computation of the residuals, the classification process is executed. For each sub time series the residual of the target variable is compared to the residuals of the historical observations. As discussed in chapter 5, this is done by developing a boxplot. The boxplot identifies large outliers based on classification boundaries. The selected technique of classifying anomalies is based on the distance of the residuals from the ’box’. This distance is expressed in number of InterQuartileRanges (IQR’s) outside the box. The classification boundaries indicate the number of IQR’s outside the box that result in an anomaly. This distance is different for both the ‘first 13 weeks’ and the ‘after 13 weeks’ sub-time series. This design choice will indicate which classification boundaries are utilized. The boxplot concept generally uses the 1.5 * IQR and 3 * IQR as boundaries for small and major errors respectively [18]. However, to give a comprehensive overview, multiple combinations of classification boundaries are evaluated in order to determine the optimal boundaries per sub-time series.

The goal of the anomaly detection is to accurately classify customer demand forecasts as anomalies.
With the selection of the classification boundaries, the sub-time series relevance is taken into account. For each time series, both sub-time-series are evaluated with their complementary classification boundaries. As discussed before the relevance of an anomaly in the first 13 weeks is the highest. Therefore, the 1-13 classification boundary is smaller or equal to the 13+ classification boundary. While adhering to this design rule, all different boundary combinations are evaluated. The $F_1$ score, the precision, the accuracy and recall are used to evaluate the boundary combinations.

### 6.2 Forecasting parameter optimization

Both forecasting methods use multiple parameters to ‘forecast’ the historical observations. These parameters largely influence the performance of the forecast. Therefore, two different techniques are used to select the best performing forecasting hyperparameters. These concepts are discussed in the following subsections.

#### 6.2.1 Grid search

To compute the optimal forecasting parameters to compute the normal behavior, a ‘grid search’ is used. A grid search is a hyperparameter optimization technique that evaluates different compositions of hyperparameters. The goal of the grid search is to find a composition of parameters that minimizes/maximizes an objective variable or function. Nexperia has 15000 time series with different behaviors, that have to be evaluated for anomalies each week. The goal of this grid search is to identify these compositions of hyperparameters for all time series. These hyperparameters should represent the original data accurately. Different objective variables are have been selected in chapter 4. The most relevant measure for our problem situation is the Mean Absolute Error (MAE), and is used as objective variable for the grid search. Mainly because it is largely seen as an effective methods that can deal with pre-existing outliers in the historic data. Both forecasting methods have different hyperparameters, and the ranges used in the grid search are discussed below. As discussed each time series is divided into two different sub-time series. For each of these sub-time series, a separate grid search is executed, because the behavior between the sub-time series can differ. This results in a grid search for 30,000 different sub-time series.

**ARIMA parameters**

As explained, ARIMA uses three hyperparameters: Autoregressive, Integrated, and Moving Average. Different ranges are selected to give a comprehensive grid search, while minimizing the MAE.

- Autoregressive range = [1,2,4,6]
- Integrated = [1,2]
- Moving Average = [0,1,2]

These parameter ranges are combined in the grid search and each time series results in a best performing hyperparameter composition. This composition is used in the execution of the anomaly detection. For the ARIMA grid search, the integrated parameter is computed separately, based on the stationary assumption. Therefore, the integrated parameter is fixed before executing the grid search. For Nexperia it is assumed that all time series have a certain trend in the data, as discussed in chapter 2. Therefore, all sub-time series are at least differenced once, and if the stationary assumption does not hold, the differenced time series are differenced again. After the computation of the Integrated parameter, the ARIMA grid search executes 12 different parameter combinations for each sub-time series.
Double exponential smoothing (DES) parameters

Double exponential smoothing can account for trend but not for seasonality. Therefore, it has only 2 parameters. The level parameter and the slope parameter. To give a comprehensive grid search, many different parameter combinations are evaluated. Similarly to the ARIMA method, the Mean Absolute Error is used to identify the most effective parameter configuration. An advantage of the Double Exponential Smoothing method, is that there are no prior assumptions besides the univariate assumption. This prevents the computation of the degree of stationary which is necessary with the ARIMA method. The used parameter ranges for the grid search are:

- Level = [0.2, 0.3, ..., 0.8]
- Slope = [0.2, 0.3, ..., 0.8]

These ranges for the parameter values combine to a total of 49 combinations. This is selected because the computation speed of the Double Exponential Smoothing is significantly faster than the computation speed of the ARIMA method. This will be discussed in more detail in section 6.4.

6.2.2 Concept drift

The result of the grid search is the optimal composition of forecasting parameters for each product-customer combination. Both sub-time series receive a separate hyperparameter compositions. This composition is relevant for the current historical behavior of the product-customer combination. However, when new data is added to the time series, the behavior of the time series can change. This could lead to a concept drift, which means that the forecasting parameters do not explain the used time series well [48]. Therefore, concept drift can be used to identify the moment that a grid search should be executed again. This research will not execute the concept drift because only snapshots of reality are evaluated. However, in literature, fixed intervals are used to compute the concept drift as well as statistical deviations. Therefore, this will be included in the recommendations for Nexperia.

6.3 Evaluation at Nexperia

In the previous chapters, the organization and method selection are described in extensive detail. In practice this anomaly detection method should be integrated in the organization. By integrating this anomaly detection method into the organization, we develop a model which collects data, executes tasks and outputs information. From this moment the practical design of the of the anomaly detection methods for the organization is addressed as the anomaly detection model.

Figure 6.5 shows a visual representation of the anomaly detection model. It is clear that only one of the activities is the actual anomaly detection execution in which the selected method of chapter 5 are used to detect anomalies. However, additional activities support and influence the anomaly detection process. Therefore these activities are included in the model design.
As can be seen in the figure two separate activities are not discussed yet; 'Management Control', and 'Employee Review'. These activities are explained in their respective subsections below.

### 6.3.1 Management control

To ensure value for Nexperia as an organization, the anomaly detection model should not cause more effort than it yields value. Therefore, demand managers should always be able to manipulate the amount of anomalies that should be checked on a weekly basis. To deliver the most value from the selected anomalies, the most critical anomalies for the planning engine should be selected. Currently, anomalies are classified based on the dissimilarity from its historical behavior. This current approach is effective for evaluating the detection method but can be improved to yield more value for Nexperia. The value could be increased by introducing an impact variable, that takes into account the impact of an anomaly on the production planning.

**Anomaly scores**

From the boxplot explained in chapter 5, anomaly scores result from the anomaly detection model. These scores are based on the distance of an observation from its historical behavior. These distances are expressed in number of Interquartile ranges outside the boxplot. These distances can be used as a likeliness that an observation is an anomaly. A higher score relates to a further distance from historical behavior, which can be assumed to have a higher likeliness to be an anomaly. This distance from the boxplot could therefore be used to represent the chance that an observation is an anomaly.

**Demand volume**

To optimize the relevance of the anomaly detection method, the detected anomalies should be reviewed in more detail. Nexperia produces multiple products which are sold to multiple customers. In the time series concept introduced before, each product-customer combination is separately analysed. However, in the production planning of Nexperia, total product demands are critical. Therefore, the demand volume of an anomaly as part of the total demand for that product should be taken into account. Identifying anomalies while taking into account the total demand of the product, can present more information about the impact of an anomaly on the planning engine. If an anomaly exists with a large part of the total demand, the impact of the anomaly on the production planning will be higher. Therefore, we recommend to use the demand volume in order to increase the relevance of the overall model design.

**Impact**

The anomaly score and demand volume can be used to construct a variable, that takes into account the
impact on the total demand, as well as the probability that an observation is an anomaly. This variable could be constructed as follows,

\[ \text{Impact} = \text{AS} \times \left( \frac{\text{DC}}{\text{D}} \right) \]

This impact score would include the anomaly score and the demand volume. This variable could be used to select the most impactful anomalies, with regard to the planning engine. Subsequently, the impact variable could be used by demand managers to select the most impactful anomalies on a weekly basis, for which reviewing capacity is available.

### 6.3.2 Employee Feedback

Currently, the customer demand forecasts for the first 13 weeks are evaluated based on the 33% change threshold. Customer Service Representatives (CSR’s) review and correct these potential anomalies. The anomaly detection model that is proposed should implement this behavior to prevent the potential anomalies from entering the planning engine. However, more value could be realised by this employee review process. From the feedback of the employees the actual performance of the model can be evaluated. One of the main issues of the problem situation was that no labeled data was available, and collecting it was expensive. However, by utilizing the feedback of the employees based on the produced potential anomaly list evaluation measures can be computed that can quantify the performance in real time. Eventually, this feedback could be used to optimize the classification boundaries of the anomaly detection model by iteratively updating the classification boundaries. Currently, fixed classification boundaries are used to classify observations as anomalies or not. The feedback of the employees could be used to transform these fixed boundaries into variable boundaries. Several decision rules could be used to ensure this flexibility of classification boundaries, based on the overall model performance.

1. if \( \text{Evaluation Measure (EM)} > \text{Threshold (} T_{\text{max}} \text{)} \):
   
   Decrease classification boundaries to search for anomalies in lower classification regions.

2. if \( \text{Evaluation Measure (EM)} > \text{Threshold (} T_{\text{min}} \text{)} \):
   
   Increase classification boundaries to search for anomalies in lower classification regions.

It is clear that the Evaluation Measure (EM) can only be computed based on the True Positives and False Positives checked by the employees. Therefore, complex measures such as \( F_1 \) score cannot be used, but more simplistic measures such as precision have to be used.

A visualization of the decision rules for 1 product-customer combination is visualized in Figure 6.6.
These decision rules are not implemented in this anomaly detection model currently, because the used dataset of 212 time series is a small subset of time series in comparison with the 15000 time series that enter the planning engine on a weekly basis. This concept of variable classification boundaries should be tested in extensive detail. Extensive testing is necessary because human employees are involved. Many research papers have indicated that when working with human feedback, the performance of the model can differ between employees [18]. Therefore, this concept could be useful to ensure robustness towards future demand pattern changes, however, when implemented it should be tested extensively.

6.4 Results

This section discusses the classification results of the anomaly detection model. These results mainly include the differences in evaluation measures between the two forecasting methods. However to take into account the practical applicability, the computational efficiency is also taken into account.

6.4.1 Training data

To evaluate the performance of the anomaly detection model, the preprocessed time series are used. All time series are divided into two sub-time series; one for the interval 1-13 weeks, and one for the weeks after the initial 13 weeks. As discussed, sub-time series with the length of 39 observations are used to classify the new observation as anomalous or not. A planning employee has classified the 11 time series as anomalous from the 212 time series. To increase the relevance of the evaluation we distinguish major and minor errors. Five major errors exist and six minor errors were identified from the 11 anomalous time series. Nexperia expressed the possibility to process anomalies on a weekly basis, however, the organization indicated that it is not effective to process hundreds anomalies on a weekly basis. In practise this will ensure that minor anomalies are not checked often, but to identify the classification performance of the model, both major and minor anomalies are evaluated in this research.
6.4.2 Classification performance

After selecting the most effective anomaly detection methods from literature in chapter 3, and determining techniques to evaluate the performance of the methods in chapter 4, we discussed the algorithm and additional design choices in the beforementioned sections. The results for both forecasting methods are discussed by evaluating multiple classification boundary combinations. It should be clear that both sub-time series are used to classify the customer-product combination as anomalous or not. Both sub-time series utilize separate classification boundaries as explained in chapter 5. The results are presented with different evaluation metrics, extracted from chapter 4. The error value is expressed in terms of IQR’s. This indicates the degree of difference between the target residual and the residuals of the historical behavior. When this difference is high, a large IQR is expected, which indicates a large possibility that the target observation is an anomaly. The following figures show the error score for all 424 sub-time series. Two main divisions are made: forecasting method division and forecasting horizon division. Both forecasting methods deliver different results and are discussed in their respective sections. For each of these methods, a division between the forecasting horizons is made. All error scores for the 1-13 week sub-time series are visualized first, whereafter, the error scores for the 13+ sub-time series are visualized. The major errors are visualized by red dots and minor errors are visualized by green dots.

Double Exponential Smoothing results

Figure 6.7 and 6.8 show the IQR distances from the boxplot for the exponential smoothing method. As can be seen in the Figure 6.7, many sub-time series receive a high error score, while not being classified as an anomaly by the planning employee. This is not desirable for the detection model. However, the detection model is a combination of the two different intervals. This means that it could be true that a product-customer combination has an anomaly in the first 13-weeks, and not in the interval afterwards (and vice-versa).

![IQR distances Exponential smoothing period 1-13 weeks](image)

Figure 6.7: IQR distances from week 1-13 with the exponential smoothing method
The overall performance of different classification boundaries is visualized in table 6.2. In this table classification boundaries [1,2,3] were selected in the first 2 columns. For these boundaries the overall model performance is computed based on the 'or' operator. If the error score is higher than the boundaries in either the first 13 weeks, or the weeks after the 13th week, the product-customer combination is classified as an anomaly.

Table 6.2: Results exponential smoothing

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Acc</th>
<th>Precision</th>
<th>Recall</th>
<th>$F_1$ Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>19</td>
<td>2</td>
<td>182</td>
<td>0.90</td>
<td>0.32</td>
<td>0.82</td>
<td>0.46</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>8</td>
<td>3</td>
<td>193</td>
<td>0.95</td>
<td>0.50</td>
<td>0.73</td>
<td>0.59</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>4</td>
<td>193</td>
<td>0.94</td>
<td>0.47</td>
<td>0.64</td>
<td>0.54</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>199</td>
<td>0.97</td>
<td>0.75</td>
<td>0.55</td>
<td><strong>0.63</strong></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>199</td>
<td>0.96</td>
<td>0.71</td>
<td>0.45</td>
<td>0.56</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>201</td>
<td>0.96</td>
<td>1.00</td>
<td>0.27</td>
<td>0.43</td>
</tr>
</tbody>
</table>

It is clear that low classification boundaries result in many True Positives, but also yield many False Positives. Additionally, the accuracy measure is a simplistic measure that evaluates the degree of correctly predicted observations. From literature we identified that an accuracy score higher than .90 is assumed high performing. However, in our research design the distribution of Positives and Negatives is extremely skewed, therefore, this accuracy measure is not the optimal measure. The actual detection performance lies in the detection of anomalies while not detecting many False Positives. Therefore, precision, recall and $F_1$ score are introduced to capture this performance. As discussed in chapter 4 this measure is the best measure to compare the predictive performance of detection models. The only disadvantage is that no benchmarks exist to classify a model as 'decent', 'high', or 'excellent' performing. Despite this disadvantage, it is still effective in comparing models, therefore the main focus lies on this measure. It is clear that the highest $F_1$ score is achieved with classification boundaries of 2 IQR for both sub-time series. Therefore, this would be the optimal classification boundaries for the Double Exponential Smoothing method if both major and minor errors would be taken into account.
Autoregressive Integrated Moving Average results

The results of the ARIMA method are presented in Figures 6.9 & 6.10.

![IQR distances ARIMA period 1-13 weeks](image1)

**Figure 6.9:** IQR distances from week 1-13 with the ARIMA method

![IQR distances ARIMA period 13+ weeks](image2)

**Figure 6.10:** IQR distances from week 13+ with the ARIMA method

Similar behavior exists when comparing the Double Exponential Smoothing method with the ARIMA method. However, it can be seen that a better distinction is made between errors and no errors. When comparing the actual classification performance from table 6.2 & 6.3, it can be seen that the ARIMA method performs better in the classification process. A maximum $F_1$ score of 0.74 is achieved in comparison with the maximum of 0.63 with the smoothing method.
Table 6.3: Results ARIMA

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>TP</th>
<th>FP</th>
<th>FN</th>
<th>TN</th>
<th>Acc</th>
<th>Precision</th>
<th>Recall</th>
<th>F₁ Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>9</td>
<td>24</td>
<td>2</td>
<td>177</td>
<td>0.88</td>
<td>0.27</td>
<td>0.82</td>
<td>0.41</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>8</td>
<td>3</td>
<td>8</td>
<td>13</td>
<td>0.92</td>
<td>0.38</td>
<td>0.73</td>
<td>0.50</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>6</td>
<td>13</td>
<td>5</td>
<td>188</td>
<td>0.92</td>
<td>0.32</td>
<td>0.55</td>
<td>0.40</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>7</td>
<td>1</td>
<td>4</td>
<td>200</td>
<td>0.98</td>
<td>0.88</td>
<td>0.64</td>
<td>0.74</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>6</td>
<td>200</td>
<td>0.97</td>
<td>0.83</td>
<td>0.45</td>
<td>0.59</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>0</td>
<td>8</td>
<td>201</td>
<td>0.96</td>
<td>1.00</td>
<td>0.27</td>
<td>0.43</td>
</tr>
</tbody>
</table>

The optimal classification boundaries for the Double Exponential Smoothing method are $2 \times \text{IQR}$ for the both sub-time series. For the ARIMA method $2 \times \text{IQR}$ was perceived the best boundary for both sub-time series as well. It should be taken into account that these classification boundaries are optimal for the used dataset of 212 time series. When executing a larger experiment, different classification boundaries could be optimal. Additionally, both major and minor anomalies are taken into account in these results. The relevance of only detecting large anomalies is discussed in the feasibility study of subsection 6.5.3.

6.5 Feasibility & validation

This section discusses the feasibility of the proposed methods and validates the practical relevance for Nexperia. Three different aspects are discussed: 'the generalizability of the method for other customers', 'the computational efficiency', and 'the volume of detected anomalies'.

6.5.1 Validation of generalizability

To illustrate whether or not the sample size is large enough to generalize this research for other time series, we use Cochrans Formula [41]. This formula is used to compute the margin of error in our research design. The degree of anomalies in the sample is important, and to illustrate the differences between taking into account all anomalies, and only major anomalies, the margin of error is computed for both situations. A confidence level of 95% is used.

- Margin of error when taking into account all errors: 2.93%
- Margin of error when taking into account only major errors: 1.88%

These statistics show that despite the small sample size, the sample size can be generalized for different time series if only statistics would be taken into account. However, this does not account for differences between customers of Nexperia. Nexperia has customers in many major industries including telecommunications and automotive. This provides a limitation for the generalizability, because it cannot be assumed that all customers have similar demand behavior. To tackle this limitation, an additional subset of time series is collected from different customers to validate the generalizability. This dataset is not completely analysed for anomalies. The main focus of this validation is on whether the error score distribution resulting from the anomaly detection model shows similarities with the original error score distribution of the training dataset. Additionally, the number of 'detected' anomalies per classification boundaries are compared in order to compare the model performance for the different subsets.

First the total number of anomalies per classification boundary are reviewed. Figure 6.4 shows the total detected number of anomalies in the 212 training time series, and the 53 test time series.
Table 6.4: Comparison error distribution test and training dataset

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound 13+</th>
<th>Training anomaly rate DES</th>
<th>Training anomaly rate ARIMA</th>
<th>Test anomaly rate DES</th>
<th>Test anomaly rate ARIMA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.13</td>
<td>0.16</td>
<td>0.21</td>
<td>0.25</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.08</td>
<td>0.10</td>
<td>0.15</td>
<td>0.19</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.07</td>
<td>0.09</td>
<td>0.13</td>
<td>0.19</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.04</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.01</td>
<td>0.01</td>
<td>0.02</td>
<td>0.04</td>
</tr>
</tbody>
</table>

It can be concluded from this table that for all training and test time series, the number of anomalies decrease with higher classification boundaries. This justifies the assumption that larger anomalies have a larger distance from the boxplot. Additionally, it can be seen that when a classification boundary of 1 is taken into account, variability is present. This variability decreases with higher anomaly boundaries and almost disappears by 2+ classification boundaries. This supports the prior conclusion that the overall anomaly detection method performs best on major anomalies instead of both minor and major anomalies.

To elaborate more on the error distributions, to compare different customer types Figures 6.11 - 6.18 show the histograms of the error distributions. All time series with an error score of 0 are excluded from the histogram to visualize the distribution of errors more clearly, otherwise the histogram would be heavily skewed around 0.
The histograms show that the different customers in the training and test set have a similar distribution of error scores throughout the forecasting process. This similarity shows the potential to generalize the method for different customers in Nexperia. Although, this similarity shows potential, this validation should be executed on a larger scale to make relevant conclusions. Additionally, the detected anomalies were not verified by planning employees, therefore, no direct classification evaluation can be executed.

### 6.5.2 Computational efficiency

As discussed before, the performance of a model is generally the main objective of an anomaly detection model. However, when evaluating large datasets with time constraints, the computational efficiency increases in relevance. Therefore, we have evaluated both forecasting methods based on their computational efficiency.

Table 6.5 shows the computation times of the grid search and the execution phase. All of these computations have been executed on a Thinkpad T490s with an Intel core i7 8th Gen processor. The ARIMA grid search utilizes 12 combinations of hyperparameters because the $d$ parameter is fixed to ensure that the stationary assumption can be adhered to. The smoothing grid search utilizes 49 different combinations of hyperparameters. Despite the lower number of hyperparameter combinations of ARIMA, the computational efficiency of the exponential smoothing method is still faster per time series.
Table 6.5: Computational efficiency in seconds when executing the grid search and execution process

<table>
<thead>
<tr>
<th>#time series</th>
<th>Grid Search (sec)</th>
<th>Execution (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ARIMA</td>
<td>Exponential Smoothing</td>
</tr>
<tr>
<td>424</td>
<td>4500</td>
<td>691</td>
</tr>
<tr>
<td>1</td>
<td>10.6</td>
<td>1.6</td>
</tr>
<tr>
<td>30000</td>
<td>318396 (88+ hrs)</td>
<td>48891 (13+ hrs)</td>
</tr>
</tbody>
</table>

The grid search is the critical factor for both forecasting methods. It can be seen that executing the ARIMA grid search takes 88 hours for 30000 time series. The Double Exponential Smoothing grid search is +/- 7 times faster than the ARIMA grid search. This duration is computed by executing the grid search on a standard i7 processor. When the grid search would be executed on a server with increased processing power, this execution time could be decreased. In addition, the grid-search does not have to be executed on a weekly basis. As discussed in subsection 6.2.2, concept drift is a technique in which statistical properties are computed of the forecasted time series. Based on these statistical properties, an assessment can be made to reiterate the grid search. A different approach would be to execute the grid search in a fixed interval (each quarter/month). Therefore, even if it takes 88 hours, it would still be executable if it had to be executed on a monthly basis. Additionally, Nexperia indicated that the actual execution of the anomaly detection (instead of the grid search) should be executed on a weekly basis, which only takes 41, and 1.5 minutes for ARIMA and DES respectively. Therefore, both methods are approachable, but significant differences exist in computation time.

6.5.3 Anomaly volume

After the generalizability and computational efficiency, the amount of anomalies that are presented should be evaluated. As indicated in the problem description of 1.1.3, Nexperia is able to use planning employees for evaluating potential anomalies. However, it is not possible to weekly evaluate hundreds of anomalies. Therefore, the total amount of expected anomalies should be evaluated to assess the practical feasibility. The initial amount of 11 anomalies from the 212 time series is not feasible, which would have lead to $15000 \times (11/212) = 778$ anomalies. Therefore, only the major anomalies are taken into account. To ensure that the method performs similar when only taking into account major anomalies, table 6.6 shows the $F_1$ scores when only major anomalies would be evaluated.

Table 6.6: Performance when excluding the minor errors anomalies

<table>
<thead>
<tr>
<th>Bound 13</th>
<th>Bound rest</th>
<th>$F_1$ score smoothing major anomalies</th>
<th>$F_1$ score ARIMA major anomalies</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.24</td>
<td>0.21</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.38</td>
<td>0.31</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.3</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0.5</td>
<td>0.62</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.62</td>
<td>0.55</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

It can be seen that the overall performance for both methods is comparable, but that the classification boundaries shift higher to achieve similar performance. This is as expected because major errors should be allocated higher error scores. To generalize for a weekly anomaly detection, this would result in $15000 \times (5/212) = 353$ potential major anomalies. According to Nexperia, 353 anomalies would still be too many anomalies to review on a weekly basis. Therefore, it could be relevant to select even higher classification boundaries, in order to reduce the
number of anomalies. Moreover, the impact variable as discussed in subsection 6.3.1 could be used to decrease the total number of anomalies flagged.

6.6 Discussion

This chapter introduced several design choices and evaluation methods applicable to the situation at Nexperia. These design choices were implemented and the results of the implementation were summarized in section 6.4. Several interesting insights were raised by comparing the forecasting methods.

The design choices were solely selected for this problem situation. Which means that these choices can differ between organizations, because the choices are generally designed for the data that is available. The time series were divided into sub-time series in order to take into account the forecast planning weeks. Additionally, the optimal number of 39 historical observations to model normal behavior were selected. The results of the anomaly detection model have shown potential in executing the anomaly detection process. When taking into account major and minor anomalies, ARIMA outperformed DES with a $F_1$ score of 0.74 over 0.63.

Finally, the feasibility of the method into the organization was researched. The current research uses a small subset of 212 time series, which would be extended to 15000 time series on a weekly basis. Questions could be raised about the generalizability of the method, therefore the anomaly detection was executed on a separate training dataset, in which the distribution of error scores was evaluated. The error rates and histograms in subsection 6.5.1 showed large potential to generalize the method for other customers of Nexperia. Especially, when detecting ‘major’ anomalies, the distributions are extremely similar. However, it cannot directly be concluded that the method is generalizable, because the detected anomalies were not evaluated with planning employees. Therefore the relevance of this validation is limited to the distribution of error scores and the number of anomalies flagged.

Additionally, the computational footprint of the model was analysed, which resulted in the assessment that both methods are feasible. However, the grid search (which is the largest computational bottleneck) of the Double Exponential Smoothing method performs +/- 7 times faster than the ARIMA method. As discussed in subsection 6.5.2, a trade-off should be made between the performance of the model and the computational efficiency. To conclude the feasibility study, the expected number of anomalies flagged a weekly basis were analysed. When only taking into account major anomalies 353 anomalies were expected to be flagged on a weekly basis, which will be too many to manually review. Therefore, the impact of the anomalies could be taken into account to select the most critical anomalies out of this final subset.
7 Conclusion & recommendations

This thesis is the base of a use case in which Nexperia will design an anomaly detection method that is able to flag high potential anomalies. This chapter first answers the research questions, and assess whether the research goal is achieved. Afterwards the limitations are discussed in order to subsequently present opportunities for future research. Finally, recommendations are presented for Nexperia, to ensure a comprehensive understanding of the developed model and future opportunities.

7.1 Research goal and questions

The first research question focused on the extraction of anomaly detection methods from literature:

1. What current anomaly detection methods exist that are suitable for time series data.

To answer this research question, a literature review was conducted. This literature review collected 15 different anomaly detection methods applicable to time series. Two main categories were identified in the literature: machine learning techniques, and forecasting techniques. Machine learning techniques proved to be effective in detecting anomalies with a low computational footprint, however, machine learning techniques were found to be heavily relying on accurate labeled training data, which is non-existent in our problem situation. Additionally, interpretability of machine learning techniques is often an issue, which resulted in the exclusion of these anomaly detection methods. Subsequently, forecasting methods were evaluated. The most effective forecasting based anomaly detection methods, were regression based anomaly detection methods. These methods use forecasting techniques to model normal behavior. This forecasted normal behavior is used to compute residuals, from which statistical characteristics are derived. Subsequently, these characteristics are used by a classification technique to classify deviating observations as anomalies. This thesis evaluated two forecasting methods to model the normal behavior: Double Exponential Smoothing (DES), and Autoregressive Integrated Moving Average (ARIMA). These forecasting methods were relevant for Nexperia’s problem situation, because both methods are able to deal with trends, and variability in historic data. ARIMA proved to be to be the most accurate forecasting method, but DES outperforms ARIMA based on computational efficiency. This trade-off is valuable when selecting the most effective anomaly detection method for the problem situation. The selected classification technique is the non-parametric boxplot method. The boxplot was selected because large errors existed in the training data, which resulted in skewed residual distributions.

The resulting anomaly detection method is implemented for both forecasting methods, in order to answer research question 2:

2. How do the selected anomaly detection methods perform in practise, and how to optimize this performance?

For this research question, the anomaly detection method is implemented for both forecasting method and the corresponding results are compared. To optimally compare the methods, multiple design choices were taken into account:

- forecast relevance
  
  First, the relevance of forecasting horizons were taken into account by dividing each time series into two sub-time series. As indicated by Nexperia, forecasts for the first 13 weeks after the Rolling Week, have the highest impact on the ability to fulfill demand. However, forecasts after the first 13 weeks are able to impact the production planning as well, but in a less impactful way. Therefore, both sub-time series are separately evaluated with their own classification boundaries.

- horizon interval
  
  Behavior of demand forecasting can change over time, and the relevance of forecasts made two
years ago was assessed. It is assumed that forecasts made in the recent past are more relevant than forecasts made long ago. Additionally, to keep computation time low, the least amount of historical observations should be used in the modeling of normal behavior. Therefore, the relevance of horizon interval is evaluated and the most relevant time series length of 39 observations is selected. This interval is selected based on the assumption that all time series should be stationary.

After executing the anomaly detection based on the training data, while taking into account the design choices. The results were assessed based on several evaluation measures extracted from a literature review. The results indicated a difference in performance, where the ARIMA forecasting method achieved a higher accuracy than the DES method. Anomaly detection with ARIMA forecasting achieves a maximum $F_1$ score of 0.74, while anomaly detection with DES achieves a maximum $F_1$ score of 0.63.

This research question also focused on the optimization of the performance, which is why a feasibility study was implemented in order to assess and improve the feasibility of the method for Nexperia. The main focus was to assess the generalizability, evaluate the computational efficiency, and discuss the practicality of the methods. The generalizability was evaluated by evaluating the error distributions of a test dataset. This evaluation showed large similarities when only the major anomalies were taken into account. This similarity was elaborated in more detail by iteratively evaluating the detection of only major anomalies in the training data set. This evaluation showed to be extremely valuable, because the DES method achieved a similar $F_1$ score as ARIMA in detecting only major anomalies. This also indicated that the total number of anomalies could be decreased to a error rate of 2% instead of 5%. Therefore the main performance optimization can be achieved by only identifying major anomalies. Moreover, Double Exponential Smoothing is the optimal forecasting based anomaly detection method for the used training dataset.

By answering both research questions, the main research goal was achieved:

**Design a tool that automatically detects potential anomalies in customer forecast data, in order to improve the data quality.**

An effective anomaly detection model is constructed that can identify potential anomalies in time series data. 'Major anomalies’ can be identified with a high performance, where ‘minor anomalies’ are more difficult to correctly classify. Two different forecasting techniques are evaluated: Autoregressive Integrated Moving Average, and Double Exponential Smoothing. In general, the ARIMA method achieves a higher performance, however, the ARIMA method has an considerable larger processing time. Therefore, when selecting the forecasting technique, a trade-off between performance and computational efficiency should be taken into account. Moreover, Nexperia stated the desire that for the future the goal is to execute the anomaly detection model on a more regular basis than weekly. This shows that it could be more effective to select the less accurate method, in order to achieve a higher processing time. To elaborate more on this effectiveness of the method for Nexperia, a feasibility study was executed in order to assess whether or not the anomaly detection model could be implemented for the entire organization. This feasibility study raised many interesting insights, that could be helpful for the organization on a short and long term basis.

The classification process is mainly dependent on the correct selection of classification boundaries. These classification boundaries select, based on the error score, what observations are classified as anomalies. However, these boundaries should be carefully selected to ensure the highest performance of the detection model. It is assumable that demand behavior changes between customers, and could change over time. Therefore, a periodic review of the model parameters is proposed to ensure a relevant and effective anomaly detection model.

Finally, the anomaly detection model can be used to flag anomalies, however, it should be taken into account that the detection model was designed for the purposes of Nexperia. The training data used is only 1.5% of the weekly data received and design choices were made based on this training data. This shows the necessity of an extensive test phase, which focuses more on the differences in forecasting behavior.
between customers. A larger and more diverse dataset should be used to test the overall performance of the model. However, despite the limitations, the anomaly detection model shows large potential to identify major anomalies in the customer demand forecasts and could be a relevant contribution to the supply chain department of Nexperia.

7.2 Limitations

Although the implemented method achieved a high performance, several limitations still exist.

7.2.1 Impact on the planning engine

One of the main limitations of this research is that the impact of an anomaly on the planning engine is difficult to define. The planning engine uses a complex linear program with multiple restrictions and objective rules. The impact of an anomaly entering this planning engine is hard to trace. Therefore, combining the anomaly detection model with the information about the data volume which is discussed in subsection 6.3.1 could improve the relevance of the automatic anomaly detection. This impact variable could help with defining, and increasing the value of an anomaly detection model for the planning engine. However, to implement this, multiple databases should be connected, which was not feasible for this research. Therefore, this will be a recommendation for future research.

7.2.2 Dataset limitations

As discussed before, the goal of Nexperia is to analyse 15000 time series on a weekly basis. However, this research only selected 3 customers, of which the data was collected. This resulted in 212 useful time series that were used to train the model. This limitation could lead to multiple issues when implementing this method for other customers. Customers can have different forecasting behavior, therefore, when implementing this method into the organization, a larger test set should be used to evaluate the classification boundaries of the model.

To account for this limitation, a feasibility study was executed where data from 2 different customers was collected. Based on the unseen data, the distribution of error scores was compared, as well as the number of raised anomalies. This feasibility study has shown that the 2 additional researched customers, have similar error distributions for both forecasting methods. Especially, when only evaluating major anomalies, the forecasting methods are extremely similar. Therefore, the tentative validation shows the potential to generalize the method for all customers. However, this should still be investigated in greater detail.

7.2.3 Cumulative series limitations

In chapter 2, the cumulative time series concept was introduced. Several advantages were stated that lead to the development of this construct. However, several limitations are discussed as well. The cumulative series removes information by taking the cumulative value of an EDI. It could also be useful to zoom more into the individual EDI’s, in order to find anomalous patterns. Therefore, in the recommendations we will propose the development of a dashboard which includes both the cumulative series (from which anomalies are computed) and the EDI’s to dive more into what caused the anomaly. This anomaly cause can have different origins, from employee errors to planning nervousness, and knowing the cause could be helpful to prevent anomalies in the future.

The forecasting horizon distinction was a major design choice that was implemented to deal with this limitation. This horizon distinction implemented the division between forecasts made for the coming 13
weeks, and forecasts made for weeks after the first 13 weeks. This distinction tries to capture information from the EDI’s about the planning horizon, because forecasts for the first 13 weeks have a larger direct impact on the production planning. However, this still remains a limitation.

7.2.4 Employee feedback

As discussed in section 6.3.2, employee feedback can be extremely helpful to ensure robustness towards the future, by adjusting the classification boundaries. Additionally, the planning employees can be used to check potential anomalies and remove actual anomalies from the data. This would lead to an improvement in the quality of the data. However, working with humans always presents implications. Different employees can have different opinions about the identity of an anomaly. This is amplified due to the absence of anomaly rules. Therefore, the involvement of planning employees to process anomalies could result in implications.

7.3 Future research

As discussed in the last section, many limitations were present when conducting this research. These limitations presented opportunities for future research and these opportunities are discussed in this section.

Anomaly impact

As discussed in chapter 6, the consequences of an anomaly entering the planning engine cannot be computed, because of the complex linear program. However, it can be assumed that when large anomalies enter the planning engine, that the accuracy of the production planning decreases. Currently, the anomaly detection model does not take into account the demand volume of the anomalies, which decreases the relevance. It can be possible that a small order is flagged as an anomaly, but that the impact on the production planning is neglectable. Therefore, the impact variable is discussed in this thesis, however it is not implemented yet. This concept could increase the value of the anomaly detection model immensely because it takes into account the effects of the anomaly on the planning engine.

Human-application interaction

This thesis discussed the techniques used when detecting anomalies and applied them on a small sample size of the data. For future research purposes, a dashboard could be developed where employees could interact with the model as discussed in chapter 6. This could lead to a research where the interaction between the anomaly detection application and human employees can be evaluated. Does the model detect the anomalies with a high accuracy and is it possible to implement the employee feedback to the detection model correctly. This future research recommendation is a large project which could be divided into the development of the application, and the human-application interaction research.

Execution frequency increase

As discussed, this thesis assumed a method that can be executed on a weekly basis. However, Nexperia expressed the desire to increase the frequency in the future to a daily execution. This would lead to many challenges for the detection model and could be used in future research. With the current problem situation, forecasting based detection was the optimal choice, however, it could be interesting to use the resulting anomalies from this model in order to train a machine learning model. From literature we have seen that ML models often perform faster and more accurate. Assuming that for future research enough labeled training data is available, ML anomaly detection could be interesting, to increase the frequency of the execution.
7.4 Recommendations

The purpose of this section is to discuss the conclusions while taking into account the limitations to present useful recommendations to Nexperia. These findings can be helpful when developing the use case for the automated anomaly detection.

7.4.1 Design choices

When designing the anomaly detection method for Nexperia several design choices should be made. The main design choices that arise are the time series length and the forecast horizon relevance. The optimal derived length for the time series in this thesis is 39 weeks. However, this is modeled purely based on the available training data. To utilize this design choice optimally, a bigger and more diverse training set should be used to select the optimal time series length throughout the entire organization. Additionally, when the DES method is selected, the stationary assumption is not present. Therefore, the classification performance for different time series lengths could be evaluated to select the optimal horizon length.

The second design choice divided all time series into two sub-time series in order to make a distinction between EDI forecasts for the first 13 weeks and the EDI forecasts after the first 13 weeks. This division should be evaluated more carefully, because we assumed that the forecasts for the coming 13 weeks have the most impact on the planning engine. Although forecasts in the near future are more impactful for the planning engine, business decisions arise to identify this relevance. It could be decided that only the forecast for the coming 26 weeks impact the planning engine, and that only those weeks are evaluated.

7.4.2 feasibility major and minor anomalies

By executing the anomaly detection model on the training dataset, several results were collected. It showed that the ARIMA method outperformed the DES method with a $F_1$ score of 0.74 and 0.63 respectively. After the feasibility study, it showed that the anomaly rate in the training data was perceived to be too high. Therefore, a distinction was made between minor and major anomalies. After evaluating the training set, while neglecting the minor anomalies, the classification performance of both forecasting methods increased. ARIMA increased slightly from 0.74 to 0.75, and DES increased the $F_1$ score from 0.63 to 0.75. This increase of performance and decrease in anomaly volume showed that it would be beneficial for Nexperia, to only review major anomalies.

7.4.3 Employee review

As indicated in chapter 1, the goal of the anomaly detection method was to be robust for future changes. One of the proposed techniques to ensure this robustness was the implementation of the employee review. This review process would serve two purposes: correcting anomalies and improving the classification boundaries. The correction of anomalies can be implemented easily by letting planning experts review the potential anomalies. If the potential anomaly is an actual anomaly, the demand forecast can be corrected. The second purpose is more complex and discussed in more detail in subsection 6.3.2. Optimizing the classification boundaries of the anomaly detection model based on the feedback of employees can produce many issues. The main issue is that when working with different human employees, the model would be altered based on their knowledge. This decrease the overall performance of the anomaly detection model when incorrect or inaccurate feedback is given to the model. Therefore, we recommend to analyse the influence of different employee behavior on the anomaly detection model in great detail. This would ensure that the employees would not impact the performance of the anomaly detection method negatively.
7.4.4 Demand volume of anomalies

From the limitations of this thesis we have seen that the main limitation is the impact of an anomaly on the planning engine. This thesis focused purely on the detection of high potential anomalies, and we concluded that to increase the value of such an anomaly detection model, the demand volume should be taken into account. If an anomaly represents a very small part of the total demand for a product, detecting that anomaly is irrelevant. Therefore, we recommend to introduce a threshold, where an anomaly should represent an x amount of percent of the total demand volume of a product to be detected. This would increase the practical relevance of the anomaly detection model because it will only detect anomalies that actually, have a significant impact on the production planning.

7.4.5 Collection of anomalies

An additional limitation of the problem situation was that currently, no knowledge was present about the behavior of anomalies. Very simplistic detection rules exist, where only the first 13 weeks of an EDI forecast are evaluated, and where only the change in total demand is taken into account. For the purpose of anomaly detection it is beneficial to have information about the actual behavior of anomalies. Therefore, we recommend to collect examples of anomalies that can be used in the future to train machine learning models. These models have often a higher performance and achieve a higher computational efficiency. However, these models generally need lots of labeled training data.

7.4.6 Dashboard

The main aggregation concept that was introduced in this thesis is the aggregation from multiple EDI forecasts to a cumulative demand forecast. This aggregation can directly be used in Nexperia’s organization in order to visualize the total demand between weeks. Visualizing cumulative changes between weeks, can be used to remove the largest anomalies present. Additionally, by visualizing trends, variability and seasonality can be identified more easily. Therefore, we recommend to utilize a Customer forecast dashboard, in which both the cumulative time series as the individual EDI forecasts are visualized. An example of such a dashboard can be seen in appendix E.
Bibliography


Appendix A visualizes variability between consecutive weeks in the EDI, which indicates that it is hard to identify a single cause of an anomaly. Many customers share EDI’s with large variability which introduces the need for a more stable concept. The figures show three consecutive weeks in 2019 where inaccurate forecasts are shared, which would not be captured by the current 33% rule.

Figure A.1: A visualization of an EDI made in week 18 of 2019

Figure A.2: A visualization of an EDI made in week 19 of 2019
Figure A.3: A visualization of an EDI made in week 20 of 2019
## Appendix B

Table B.1: Exclusion and Inclusion of variables in the training dataset

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Type</th>
<th>Included/Excluded</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUSINESS_MODEL_IDENT</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>CPFE_MESS_RECEIVER</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>CUSTOMER_PART_NUMBER</td>
<td>String</td>
<td>Included</td>
<td>Used for IDcode construction</td>
</tr>
<tr>
<td>DATE_FORECAST_BUCKET</td>
<td>Date</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>DATE_MESSAGE_MODIFY</td>
<td>Date</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>DATE_MESSAGE_RECEIVE</td>
<td>Date</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>FC_QTY_BY_CUSTOMER</td>
<td>Integer</td>
<td>Excluded</td>
<td>the focus lies with forecasts that entered the planning engine</td>
</tr>
<tr>
<td>FC_QTY_BY_NXP</td>
<td>Integer</td>
<td>Included</td>
<td>These forecasts entered the planning engine</td>
</tr>
<tr>
<td>INDICATOR_SEND_I2</td>
<td>-</td>
<td>Excluded</td>
<td>Will use Customer Part Number(CPN)</td>
</tr>
<tr>
<td>MATERIAL</td>
<td>Integer</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>PLANNER_CODE</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>PLANNING_MONTH</td>
<td>Date</td>
<td>Excluded</td>
<td>Information captured in PLANNING_WEEK</td>
</tr>
<tr>
<td>PLANNING_WEEK</td>
<td>Date</td>
<td>Included</td>
<td>Indicates target week</td>
</tr>
<tr>
<td>RECEIVING_PLANT_OF_C</td>
<td>-</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>ROLLING_MONTH</td>
<td>Date</td>
<td>Excluded</td>
<td>Information captured in ROLLING_WEEK</td>
</tr>
<tr>
<td>ROLLING_WEEK</td>
<td>Date</td>
<td>Included</td>
<td>Indicates week in which the forecast is received</td>
</tr>
<tr>
<td>SALES_ORGANIZATION</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>SALES_UNIT</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>SELLER</td>
<td>String</td>
<td>Excluded</td>
<td></td>
</tr>
<tr>
<td>SHIP_TO</td>
<td>String</td>
<td>Included</td>
<td>Used in the construction of the IDcode</td>
</tr>
<tr>
<td>SOLD_TO</td>
<td>String</td>
<td>Excluded</td>
<td>Only will take into account shipping locations</td>
</tr>
</tbody>
</table>
C Appendix C
Mathematical explanation of ARIMA and Double Exponential smoothing

C.1 ARIMA explanation

As explained in the thesis the Autoregressive Integrated Moving Average consists of 3 parameters; the AR, I and MA. In this appendix the three parameters are explained and expressed in formula’s. Afterwards, the formulas are applied on a time-series example in order to explain the process of modeling normal behavior with ARIMA.

**Autoregression**

The autoregression part of ARIMA takes into account the number of historical observation that receive the highest weight in the computation of the forecast. The following equation is used to execute the regression process.

\[ X_t = c + \sum_{i=1}^{p} \varphi_i X_{t-i} + \varepsilon_t \]

**Integrated**

The integrated parameter is directly connected to the stationary assumption of ARIMA. This parameter indicates the number of times the time-series is differenced, which means the difference between the observations and its prior observation is computed. I(1) can be mathematically represented by:

\[ y'_t = y_t - y_{t-1} \]

This can be executed to remove the trend of a time-series and the result will be a more stationary time-series. Differencing can be applied multiple times, however in general differencing more than twice does not remove additional trend from the data.

**Moving Average**

The moving average parameter takes into account the error terms of the last forecasted values and uses these to forecast the next value. The general formula for the moving average is visualized below:

\[ Y_t = \omega + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} + e_t \]

For example if only the forecasting error of the last period would be used, the MA(1) model would be used. The formula for a MA(1) model can be described as:

\[ Y_t = \omega + \theta e_{t-1} + e_t \]

The new forecast would be computed based on the lagged error of the last forecast \( e_{t-1} \), a constant \( \omega \), the diminishing parameter \( \theta \), and an error term.

**ARIMA**

The combined computation of these three separate parameters can be visualized by:

\[ r_t = \varphi_0 + \sum_{i=1}^{p} \varphi_i r_{t-i} + \alpha_t - \sum_{i=1}^{q} \theta_i a_{t-i} \]

where

- \( p \) is the number of autoregressive terms
- \( d \) is the number of non-seasonal differences
- \( q \) is the number of lagged forecast errors
- \( \varphi \) is the autoregressive constant
- \( \theta \) is the moving average constant
- \( t \) is the number of time series data items
r is the forecast value
a is the moving average value.

This formula is implemented in python 3.7 by utilizing the statsmodel package.

**Example computation**

An example computation where the hyperparameters are (2,1,0) is shown below:

$$(r_t - r_{t-1}) = \varphi_0 + \varphi_1 \times (r_{t-1} - r_{t-2}) + \varphi_2 \times (r_{t-2} - r_{t-3})$$

the $r_t$ represent the forecasted values of the previous steps, and the $\varphi$ are constants optimized by the python package.
Below potential anomalies in the training data are visualized. It is assumed that the large spikes, are anomalies because the spikes are corrected in the consequent week(s).
E Appendix E

This appendix shows an example dashboard which could be used to visualize both the cumulative time series as the EDI forecasts in order to visualize the information from both time series concepts.

Figure E.1: A example dashboard including the cumulative time series as the EDI forecasts.