MASTER

Throughput time and time window estimation for business processes using historical data

Peters, S.P.F.

Award date:
2016

Link to publication
Throughput time and time window estimation for business processes using historical data

by:
S.P.F. (Sander) Peters
Student number: 0741926

Supervisors:

- dr. ir. R.M. (Remco) Dijkman - TU/e IE&IS Information Systems Group
- dr. ir. B.F. (Boudewijn) van Dongen - TU/e M&CS Architecture of Information Systems Group
- prof. dr. ir. P.W.P.J. (Paul) Grefen - TU/e IE&IS Information Systems Group

in partial fulfilment of the requirement for the degree of:

Master of Science in Operations, Management and Logistics
Master of Science in Business Information Systems

Eindhoven, Wednesday 28th September, 2016
TUE. School of Industrial Engineering
Series Master Theses Operations Management and Logistics

TUE. Department of Computer Science
Series Master Theses Business Information Systems

Subject headings: Throughput Time Prediction, Time Window Estimation, Empirical Distribution
Abstract

In this thesis a new technique for throughput time estimation and time window estimation is presented. Throughput time prediction is important for companies to plan resources or provide customers with an estimated completion time of their case. In order to increase the reliability of these estimates the new technique generates empirical probability distributions that describe when each event in the process occurs, based upon historical data. These empirical distributions are combined into a new empirical distribution to estimate the throughput time for a certain event. Using these empirical distributions time window estimates are derived with a certain precision.

Based upon the related work in the field of throughput time prediction a gap in literature on statistical techniques for throughput time prediction is found. The gap consists of the lack of the ability to use statistical techniques to obtain time window estimates. The new technique is first evaluated on an artificially generated dataset and outperforms the existing techniques for this dataset. A case study at Van Opzeeland is performed to evaluate the new technique on processes, in which there is a single possible execution path without choices, on a real-life dataset. The throughput time estimates are about 50 percent improved compared to the existing techniques and in 70 percent of the cases the time range wherein a certain even is finished is smaller with the same precision compared to the existing techniques. In order to also handle processes in which choices can be made an extension has been developed to the new technique. This extension uses the probabilities of each variant to estimate throughput times and provide estimates for time windows. The extension has been verified on a case study using the real-life dataset from the BPI 2015 challenge. For the case study the new technique performs less well than the average value for the throughput time estimates. Also the time range is larger than the average value with a markup value added to it. This might be an issue of the technique or could be caused by the large variation in the throughput times in the dataset. The new technique provides statistically reliable time windows and therefore can improve, for example, resource scheduling, since reliable estimates provide indicators when a resource is needed for a specific case.
Management Summary

When the business process is started for a customer, the customer will likely want to know when the work will be completed for him or her. At the same time the customer and the company itself may want to know when various tasks will be started for the customer. This information can, for example, be used to plan the resources that have to work on the tasks. To provide this information, throughput time estimates as well as time windows estimates need to be provided.

Based upon an overview of the related work in the field of throughput time prediction, there are many techniques developed for the prediction of throughput time. The techniques can be divided into techniques which focus on routing or on throughput time, techniques which are deterministic or statistical and techniques which are static or dynamic. From this overview can be concluded that there is a lack of statistical based techniques which provide throughput time and time window estimates.

From the related work two techniques are chosen, one focusing on throughput time prediction and one more focused on routing. The first technique is the FSM miner and analyzer (Van der Aalst, Schonenber, & Song, 2011). This technique mainly focusses on routing and provides average throughput time estimates. The second technique uses stochastic petri-nets (Rogge-Solti & Weske, 2013) to fit a distribution for each event in a process. This technique mainly focusses on the prediction of the distributions of the throughput time. Both techniques do not present a method to derive time window estimations.

The new technique uses historical data to generate empirical distributions for each event within the process. Two empirical distributions are added together by first dividing each of the two empirical distributions into segments with equal probability. A segment has a start value, an end value and a probability. Within each segment all values are assumed to be discrete uniformly distributed. The probability of each value within a segment is added to the value of a segment of the other event and their probabilities are multiplied. After all segments from both events are added to each other, a new empirical distribution is generated from the obtained values. This new empirical distribution represents the throughput time distribution for both events.

In order to obtain the expected time for the combined execution of both events the value at the cumulative probability of 0.5 is derived. Next to the expected time for the completion of both events a confidence interval or so called time window can be constructed. A time window has a certain precision and error bounds on the lower an upper part of the time window. The lower error bound marks the earliness value for the time window; this is the error that an event is finished earlier than the time window. The upper error bound marks the lateness value for the time window; this is the error that an event is finished later than the time window. The time window is obtained by taking the time values at the error bound probabilities from both sides of the cumulative distribution.
Next to the two techniques from literature and the new technique also a naïve technique is used for comparison of the prediction of the throughput time. The naïve technique uses the average value as prediction for the throughput time and a standard markup value for the generation of time windows.

In order to evaluate the techniques an artificially generated dataset is used. The techniques are compared on the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). The new technique performs about 50 percent better on the MAE compared to the other three techniques and about 35 percent better on the RMSE. For the time window estimation the new technique provides smaller time windows for all events with the same precision as the naïve method. Therefore the new technique is an improvement over the other techniques presented.

In order to show the performance of the new technique for strictly linear processes a case study is performed with the real-life dataset from Van Opzeeland. Van Opzeeland is a distribution company, delivering cargo to customers using trucks. In this case study the throughput time and time window estimates are calculated for several trips. The new technique performs about 50 percent better than the other techniques on the MAE for throughput time estimation. For the RMSE the new technique also outperforms the other techniques. The time window estimation using the new technique generates in 70 percent of the cases a smaller time window than the technique applied by Van Opzeeland. Overall the new technique provides a good performance for the case study.

Since the technique only focused on only strictly linear processes an extension has been made to also incorporate routing within processes. In order to handle routing all different variants of a process are identified and their respective probabilities of occurrence are determined. For each variant the estimated throughput time and time windows are generated using the new technique. The estimates for the different variants are added together proportional to their probability of occurrence to obtain the overall estimates for the throughput time and time windows.

The extension of the technique is evaluated on the real-life dataset from the BPI 2015 challenge. Compared to the naïve technique the new technique performs worse and in only 17 percent of the times the time windows are smaller than the time windows predicted by the naïve technique. This worse performance might be caused by the large variation in throughput times in the dataset.

By using the new technique reliable throughput time and time window estimates are obtained for strictly linear processes. For processes with routing more research should be done to verify whether the new technique has problems handling routing or that the new technique has problems handling large variance in the data or both. The new technique can be applied in future research for resource scheduling or time window predictions for customers.
Preface

This thesis is the end result of six month of intensive work and research and also marks the end of both my master studies. This result would not been able without the help and support of many people.

First of all, I would like to thank Remco Dijkman and Boudewijn van Dongen for supervising me along the project. During our meetings I received a lot of constructive feedback and support to point me into the right direction where needed. It was a pleasure to work under your supervision to this end result. Also I would like to thank Paul Grefen as the third supervisor for my thesis.

I also want to thank my colleagues in the Information Systems group in the School of Industrial Engineering for the great time during the work on my thesis. You really made me feel part of the group, especially my office mate Shaya, thank you for the company in Doğa.

Very special thanks to my parents, Hans and Wilma, my brother Jorrit and sister Janne. Your unconditional support has helped me through the tough moments during the writing of this thesis. Without your support this end result would not be the same. And last but not least, thanks to all my friends and especially Nik(e)s for making this an amazing time.

Enjoy reading!

Sander Peters

28th September 2016
# Table of Contents

Abstract ............................................................................................................................. iii
Management Summary ....................................................................................................... iv
Preface ............................................................................................................................... vi
Table of Contents ............................................................................................................... vii
List of Figures ................................................................................................................... ix
List of Tables .................................................................................................................... ix

1 Introduction .................................................................................................................. 1
   1.1 Research context ..................................................................................................... 1
   1.2 Research questions ............................................................................................... 2
   1.3 Research design ..................................................................................................... 2
   1.4 Thesis structure ..................................................................................................... 5

2 Existing Techniques for Throughput Time Prediction ............................................... 6
   2.1 Aspects .................................................................................................................. 6
   2.2 Papers ................................................................................................................... 7
   2.3 Techniques selected .............................................................................................. 13

3 Techniques for Throughput Time Prediction ............................................................. 14
   3.1 FSM miner & analyser .......................................................................................... 14
   3.2 Stochastic petri-nets ........................................................................................... 15
   3.3 Historical data based throughput time prediction (HD-TTP) ............................... 17

4 Evaluation of the Techniques on Generated Data .................................................... 20

5 Evaluation of the Techniques on Real Data ............................................................... 24
   5.1 Process description ............................................................................................... 24
   5.2 Dataset overview .................................................................................................. 26
      5.2.1 Data properties ............................................................................................... 26
      5.2.2 Data cleaning ................................................................................................. 26
      5.2.3 Data preparation ............................................................................................ 27
   5.3 Data properties ...................................................................................................... 29
      5.3.1 Legs and travel times ..................................................................................... 29
      5.3.2 Unloading times ............................................................................................. 31
5.3.3 Time windows .................................................................................................................................... 34
5.4 Application of techniques for Van Opzeeland ......................................................................................... 37
5.4.1 Techniques from literature .................................................................................................................... 38
5.4.2 Performance of the techniques ............................................................................................................. 39
5.5 Limitations of the case study .................................................................................................................... 41
5.6 Future research on the case study ........................................................................................................... 41
6 Routing Extension of the HD-TTP Technique ............................................................................................ 42
7 Evaluation of the Extension on Real Data ..................................................................................................... 44
7.1 Limitations of the case study .................................................................................................................... 46
7.2 Future research on the case study ........................................................................................................... 46
8 Conclusion ................................................................................................................................................ 47
8.1 Research question .................................................................................................................................... 47
8.2 Limitations .............................................................................................................................................. 48
8.3 Future research ....................................................................................................................................... 48
Bibliography ............................................................................................................................................... 49
Appendix A – Table overview ....................................................................................................................... 51
Appendix B – Average and Std. Dev. Stop times ............................................................................................ 53
List of Figures
Figure 1 – Example process ........................................................................................................... 1
Figure 2 – Research Design ........................................................................................................... 3
Figure 3 – Example transition system based on multi-sets ............................................................. 15
Figure 4 – Cumulative empirical distribution for event A .............................................................. 17
Figure 5 – Cumulative empirical distribution for event B (break x-axis at 40 to 80) .................... 18
Figure 6 – Combined distribution for the events A and B .............................................................. 19
Figure 7 – Example process for evaluation of the techniques ......................................................... 20
Figure 8 – Absolute average difference per activity in arrival time per technique ..................... 21
Figure 9 – Performance of the naïve window ................................................................................. 22
Figure 10 – Window Size HD-TTP ................................................................................................ 23
Figure 11 – Process model Van Opzeeland in BPMN ................................................................. 24
Figure 12 – Example route number 1484000 in the center of Rotterdam .................................. 25
Figure 13 – Frequencies of transportation legs ............................................................................ 29
Figure 14 – Aggregated frequencies of transportation legs ........................................................... 30
Figure 15 – Frequency of travel times .......................................................................................... 30
Figure 16 – Frequencies of Stops for locations ............................................................................ 31
Figure 17 – Duration of Stops in minutes ..................................................................................... 32
Figure 18 – Distributions of Stop Times per Category ................................................................. 33
Figure 19 – Aggregated Frequencies of Stops for each Client Code ........................................... 33
Figure 20 – Frequency of Stops per Category ............................................................................. 34
Figure 21 – Difference between estimated arrival time and actual arrival time distribution .......... 35
Figure 22 – Two hour window based on estimates from Van Opzeeland ................................... 36
Figure 23 – Two hour window balanced based on estimates from Van Opzeeland ................... 36
Figure 24 – One hour window based on estimates from Van Opzeeland ................................... 37
Figure 25 – One hour window balanced upon estimates from Van Opzeeland ......................... 37
Figure 26 – Absolute average difference per order in arrival time per technique ...................... 39
Figure 27 – Window sizes obtained using the new technique ....................................................... 40
Figure 28 – Performance Naïve Time window for all variants selected ..................................... 44
Figure 29 – Window size obtained using the HD-TTP technique ................................................. 45
Figure 30 – Difference estimated throughput times ...................................................................... 46

List of Tables
Table 1 – Overview of the related work analysis ......................................................................... 12
Table 2 – Example process log of a process .............................................................................. 14
Table 3 – Process activity overview .......................................................................................... 20
Table 4 – Categorized client codes for Van Opzeeland .............................................................. 28
Table 5 – Overview of Categories with their average and standard deviation stop time in minutes 32
I Introduction

In this thesis techniques on throughput time prediction and time window estimation for business processes are analyzed and a new technique is developed to incorporate stochastic methods. Stochastic methods are important to use in order to capture the variance and distributions within a process. In this chapter the research context, research questions and research design are discussed.

1.1 Research context

This research takes place in the context of business process management and process analysis. A process is defined as: “a usually fixed or ordered series of actions or events leading to a result” (Meriam-Webster, 2016). Each process has a start and an end time, the difference between those times is the throughput time of the process as a whole. The different events in a process also have their own start and end time and therefore also a throughput time for each event. Each event can require resources to execute the event. Resources are limited to a certain capacity; when the capacity is reached new cases need to wait for the resource to become available, the time until the event can be executed from the arrival of the case at the event is called the waiting time. A case is a process instance of a process executed for a specific customer. In order to illustrate this in Figure 1 an example business process is provided with several events. Each of these events requires a resource and are executed in the order indicated by the flow arrows.

Figure 1 – Example process

Strictly linear processes have one pre-defined path throughout the process, so when the business process is started for a customer the path through the process is known. For the example process it already is known that a simple claim filed will follow the path Receive Claim, Register Simple Claim and Payout Claim. Routing processes do not have one pre-defined path, but multiple possible paths a
customer can take each with its own probability. So instead of the strictly linear path a claim is assessed when the claim is received and the routing for the claim is based upon this assessment. This adjustment of the routing in the process is made during process execution. An adjustment during process execution can be defined as an adjustment while a case is already initiated, but has not yet finished. Each possible different path is called a variant of the process.

When the business process is started for a customer, the customer will likely want to know when the work will be completed for him or her. At the same time the customer and the company itself may want to know when various tasks will be started for the customer. This information can, for example, be used to plan the resources that have to work on the tasks. To provide this information, throughput time estimates as well as time window estimates can be provided. A time window is defined as a time range wherein a certain event is finished. These estimates can be generated using naïve methods, based upon averages and static values, but also statistical based methods can be used. These statistical methods use statistical probability distributions or statistical techniques in order to generate the estimates for the throughput time and time windows.

1.2 Research questions
The goal of this research is to develop a new technique for the prediction of reliable throughput time and time window estimates for business processes, based on empirical data sources. This new technique should provide accurate and reliable estimates as compared to the real data for throughput times and generate time windows with a reliable precision. From this research goal the main research question can be derived. The main research question in this thesis is:

How can reliable and statistics based estimates for throughput times and time windows be generated?

In order to answer this main research question four sub-research questions need to be answered:

Which methods are currently available for estimation of throughput time and time windows?

To what extend are process time estimations based on statistical methods and distributions?

How can statistical methods be used in a tool to estimate throughput times and time windows?

How does the implemented tool perform against the currently available methods?

The scope of this research is limited to the prediction of throughput times and the generation of time windows for the process. Resource planning and the discovery of the process are out of scope for this research.

1.3 Research design
The research design which is used to answer the research questions are discussed in this section. This research design is based on the stream of Design Science according to Hevner et al. (2004). The research design is visually shown in Figure 2.
First a literature study is performed to review the related work on throughput time estimation. Each technique in the literature is reviewed on three aspects and two techniques are chosen to compare the new technique to. Also a possible gap in the literature is indicated in this chapter. This answers the first two sub-research questions.
Next the procedures for both selected techniques are discussed step by step. These descriptions provide clear insight into how the different techniques work and how the throughput times are estimated.

After the description of the existing techniques the new technique is introduced. This technique is based upon the derivation of empirical distributions from historical data. In order to calculate the throughput time until a certain event has finished, multiple empirical distributions need to be added together. The method to do so is also part of the new technique developed. This process is an iterative process with the performance of the technique, this in order to optimize the technique. This answers the third sub-research question.

The performance of the technique is measured on an artificial generated dataset, which is used as an example for a business process. For the two selected techniques from the literature, the new technique and the naïve technique the estimated throughput times are compared. Also the estimations for the time windows are made by first determining the precision of the time window and using the new technique to obtain time windows with an equal precision, but with a smaller span. This answers the fourth sub-research question, in combination with the case studies performed.

In order to show that the new technique works for strictly linear processes in real-life, a dataset from Van Opzeeland is used. Van Opzeeland is a distribution company where trucks deliver cargo to customers. First data cleaning and preparation is performed to make the data usable for a performance analysis for the different techniques. The properties of this dataset are also provided in order to provide more insight into the dataset. The comparison on throughput time is made between the different techniques and the time window size is compared with the actual time windows from Van Opzeeland. The implications made by the new technique are reflected via this case study.

Since the technique is implemented for only strictly linear processes and extension is made to incorporate also processes where routing is applied. In order to do so a new method for handling the different variants of the process is introduced. This new method uses the previous presented new technique to calculate the throughput time estimates for each variant. The variants are combined to provide an estimated throughput time and time window by probability.

To provide how the extension of the technique performs a real-life dataset is used from the BPI 2015 challenge is used. The estimated throughput times are compared to the naïve technique and also the time windows are compared to obtain the performance of the extension of the technique.
1.4 Thesis structure

Figure 2 also presents the structure of the remainder of this thesis. In chapter 2 a cross-section of the literature on throughput time estimation is discussed and a linear and routing technique are selected. In chapter 3 the selected techniques are explained and the new technique is introduced. Chapter 4 compares the different techniques to each other on an artificially generated dataset. Next in chapter 5 a case study for strictly linear processes is performed using the dataset from Van Opzeeland. In chapter 6 an extension of the new technique is presented. This extension is evaluated using a case study in chapter 7. After all research steps are discussed an overall conclusion and answer to the main research question is provided. Also limitations of this report are indicated and directions for future research are provided.
2 Existing Techniques for Throughput Time Prediction

In this section a cross-analysis of related work on the field of techniques for throughput time prediction is presented. From the related work one technique on linear processes is selected and another technique on routing is to compare against each other. The related work is found using the literature study from Tim Claesens (2015), which provide a good overview of the available literature on throughput time perdition. Based on three aspects all techniques are discussed.

2.1 Aspects

In order to compare the different techniques three different aspects are taken into account. These are based on the different characteristics which define the properties and prediction ability of the techniques.

The first aspect focus on routing versus the focus on throughput time, some articles focus more on the routing of a process instance and less on the properties of the throughput time, and others focus more on the throughput time than on the routing of a process instance. Throughput time is the aspect for which the focus is on the remaining time needed to complete the order and mainly focus on strictly linear processes. Routing is the aspect of the different process instances and the order of process steps within these instances and where they differ from each other. A process instance is an execution of a certain process which can deviate from the original process. The routing mainly focusses on these deviations from the process.

The second aspect is static versus dynamic; this depends on how adaptive the technique is to changes in schedules or planning. The definition of static versus dynamic is adopted from the network theory according to Perry-Smith and Shalley (2003), for which static is a fixed system, whereas dynamic is an adaptable system. This difference between fixed and adaptable is checked for the techniques discussed in the papers, in terms of whether the technique can adapt at runtime.

The third aspect is stochastic versus deterministic; this focuses on the way stochastics work in the technique, whether statistical measures are taken into account or whether only deterministic methods are used. Stochastics in processes can be defined as in Gardiner (1985) as systems which evolve probabilistically in time. Deterministic properties can be defined as in Eckberg (1979) as just a single value provided to determine throughput time. Deterministic systems only work with the average value for example, where stochastic systems use statistical distributions. There are also systems which are in between both categories; they use stochastic techniques to obtain deterministic values, this category is denoted as stochastic to deterministic, since mainly calculations are made using stochastic methods, but only a deterministic value is extracted to use in further calculations.
Each of the papers from the literature study is discussed here based on the aspects described above. Based on these descriptions a technique will ultimately be chosen. The overall results of the related work analysis are presented in a table after the discussion of all papers. In Table 1 the properties of each technique presented in the different papers.

**Sha, Hsu (2004) – Due-date assignment in wafer fabrication using artificial neural networks**

This article describes the use of an artificial neural network to predict due-date assignments. The techniques used describe a production situation for which orders follow a certain routing and queue for certain servers. This method is mainly focused on the throughput time which is needed for each step and to add it all together. The setup is quite dynamic in the sense that when the setup of the production changes it can easily adapt to the new situation. The formulas are based on input parameters which are deterministic, and therefore the due date is the result of the arrival time added with the production time and the queue time.

**Chen (2010) – Fuzzy technology in advanced manufacturing systems**

Using a self-organization map fuzzy back propagation network in this article the cycle time and remaining cycle time are computed. Within this method also priorities and queuing is taken into account. This method is quite focused on throughput times and queuing within a factory setting. Furthermore the model is very static, since a change in parameters requires a complete retrain of the model. With respect to statistical measures in place, the input parameters are using statistical distributions, but no statistical distributions are provided for the remaining cycle time. So the article starts off with statistical measures, but ends up with only discrete values.

**Chen (2012) - A job-classifying and data-mining approach for estimating job cycle time**

This article uses a random classification of jobs first to later apply a fuzzy back propagation network to determine the remaining cycle time based on the size of the job, the average factory utilization, the total length of the queue at the processing route, the total queue length for bottlenecks at release, the work in progress in the factory and the average delay for recent finished jobs. For this fuzzy back propagation network an expert is needed to set the weights for the different parameters. The method focuses on the throughput time, but takes the routing for each order into account. The method proposed uses only estimates and therefore is rather deterministic, but as the estimates are made from the moment the order enters the factory it makes the model very dynamic.

**Chang, Liao (2006) - Combining SOM and fuzzy rule base for flow time prediction**

Again a fuzzy system is used here to predict cycle times within a semiconductor industry situation. Based on the current state of the shop floor a job is classified into a certain group, based on this classification using a fuzzy rule the time prediction is made. This method mainly focuses on
throughput time and has less routing properties. The throughput time calculation is based on a simulation model using the classification and fuzzy rule base. In terms of statistics this article is more deterministic then relying on statistical distributions, it yield estimates based on the order quantities, total jobs in the system, average shop workload, average queuing length, total queue in the workstation and the utilization of the bottleneck machine. With respect to dynamic adjustments of the system the classifier is trained with a static set of input parameters, which makes the whole simulation very static instead of dynamic.

_Tirkel_ (2013) – *Forecasting flow time in semiconductor manufacturing using KDD*

A decision tree model is used here in combination with a neural network classifier to identify the flow time of a job. This time is estimated based on the departure time of the previous step, the transportation time, the queuing time and the process time to calculate the departure time. Based on a classification the data is placed in a bin with the expected flow time, for which the bin size is determined by the feasible time for the factory. This method focuses heavily on throughput time and less on routing. It is a technique which provides a deterministic flow time based on the decision tree model. Since this model is only trained at the beginning and not updated in between the model is not quite dynamic, but quite static instead.

_Backus et al._ (2006) – *Factory Cycle-Time Prediction With a Data-Mining Approach*

By means of combining techniques this article tries to take the best of both worlds of clustering and a decision tree. Based on the type of lot the lots are clustered together, and instead of taking the mean of the cluster a decision tree is applied to determine the remaining cycle time. This tree should be updated if the characteristics or the properties of a factory change. This method focuses on throughput time and uses historical data to predict new cycle times. No statistical distributions are derived and, the production times are assumed to be constant. The model therefore is quite deterministic. With respect to dynamicity it is possible to adapt the model to changing data by changing the clustering or the decision tree, but this requires recompilation and needs to be done when values shift, which requires monitoring.

_Meidan et al._ (2011) – *Cycle-Time key factor identification and prediction using ML and DM*

This article does not estimate the cycle time itself, but tries to identify the main key factors which influence the cycle time. This might be very useful in complex factory setups, but is less important with respect to the calculation of the cycle time, since it does not calculate the actual cycle time. It is mainly focused on the waiting time than on the throughput time. It is a deterministic and static approach since no adaptations can be made after running the model, only rerunning the whole analysis would result in possible new data and insights.
Instead of clustering of data this article starts from a process model which is defined in BPMN, based on this model and its mathematical counterpart a routing can be established. Based on this routing the model created is used to determine the total cycle time. The cycle time is estimated using an M/Hn/1 queuing system. Both distributions are exponential, whereas the second is hyper-exponential. Based on these queuing models the service time of the different steps in the process is calculated. This model focus both on routing and throughput time and is quite statistically based using the both exponential distributions. But due to assuming the exponential distributions it limits itself on the other hand. Next to that it is a quite dynamic model, which can be applied to any business process.

Ha, Reijers, Bae, Bae (2006) - An Approximate Analysis of Expected Cycle Time

This article also relies on a process model to come up with an expected cycle time during process execution. This method uses the utilization of the agents in the business combined with their expected cycle times and correlations to come up with the expected cycle time. First the cycle time per agent per task is calculated and afterwards summed up based on the type of queuing system, assuming a steady state system. For a sequence it is added to each other, for repetitive blocks it is a multiplicity of the same block and for parallel blocks the maximum of the two blocks is taken. This method focus both on routing as on throughput time, whereas it is deterministic since it only uses the average cycle time to calculate. It is dynamic, since when something changes in the factory setting it can be easily adopted into the method.

Van Dongen, Crooy, Van der Aalst (2008) – Cycle time prediction: When will this case …

Using a non-parametric regression this article focuses on predicting the average cycle time better based on improved determination of the routing of a case and the delays which are associated with it. Non-parametric regression does not make assumptions for the function of the regression model; this type is used because cycle times are not to be assumed from a certain distribution or a special form. There are several methods of estimation discussed, as is the average, an activity occurrence estimator, an activity duration estimator and a case attribute estimator. The last three are regression type estimators, which all use different properties to estimate the average cycle time. All these three are combined to use as a predictor for the average cycle time. This method is mainly focused on routing and the average time of the activities on that process log, less focus is on the throughput time. In terms of statistical properties only the average is used, and therefore the model is quite deterministic, but still uses the real times from the data. Next to that the model is reasonably dynamic since it can learn easily from previous executions and adapt to that.

Bevacqua et al. (2014) – A data-driven prediction framework for analyzing and monitoring

In this article a clustering method is used to predict the cycle time of an order. This method uses certain properties of an order to predict the cycle time. Next to the prediction a structure analysis is
performed to identify patterns within the model. Based on these patterns the clustering algorithm is enhanced. Based on these knowledge about the data derived a time prediction is made, two times are predicted, which are the remaining processing time and the number of remaining steps. These calculations are based on the time elapsed and the category where they are put into. This method is mainly focused on throughput time, but does not work with any statistical distributions, only with the average time predicted per cluster. It is not very dynamic, because after training of the model it is harder to adapt to new changes in the data.

*Van der Aalst* (2010)– *Business Process Simulation Revisited*

Simulation is one of the major tools within business process management. This article discusses what the most common short falls of this method are and where to improve. With respect to throughput time prediction and routing this article emphasize that simulation can play a role here. It can determine what might be the next step in the process or what the expected remaining process time is according to the simulation model. This short term operation support can be used to determine the remaining throughput time of a case and therefore can also make some predictions about when a case can be finished. Statistically speaking simulation can provide confidence intervals since a simulation can be executed several times. With respect to dynamicity it is easy to change some parameters in the simulation model and rerun the model to obtain new predictions, but if structural changes take place within the simulation model need to be redesigned.

*Rogge-Solti, Weske* (2013)– *Prediction of remaining service execution time using …*

This article uses stochastic petri nets combined with arbitrary firing delays in them to predict the remaining service time of a task. This remaining service time of a task is based on the fact that there is an initial distribution for a task to be executed. Once a task has just started this distribution applies, but after time passes the distribution of the remaining service time will shift due to the fact that certain cases which were already finished are not viable predictors anymore for the real end of the current task. Based on this method the distribution shifts until it reaches the end of the distribution, when arrived there it is assumed that the case will finish immediately. The distribution needs to shift, since the area under the probability distribution always needs to sum up to one, therefore the adjustment is made and other probabilities are provided regarding the time the task ends. The method mainly focuses on throughput time and less on routing. Statistically this is a very sophisticated technique, since it uses several properties of statically distributions and adapting them throughout the duration of the task. Also the method is very dynamic since it relies upon all cases until that point to provide the estimate, which makes it easy to adapt to new data. Only when the process model differs the petri net should be adapted to fit the process model again.
Van der Aalst, Schonenberg, Song (2011) - Time prediction based on process mining

The future of a case performed in a system is not always determined upfront, therefore the time when an order is finished is hard to predict. Based on process mining techniques from event logs generated by the process a transition network is derived. The times in the log are used to calculate the remaining time for a case by using the previous cases. By assuming the law of large numbers, so larger than thirty, therefore mirroring a normal distribution, one can also compute confidence intervals. This method is mainly focused on routing, less on throughput time. The time prediction however shows that the method also takes into account the throughput time, but it is only partially the focus. Statistically speaking it only takes into account the average and standard deviation, under the assumption that those values are normally distributed. Based on this normal distribution the confidence intervals are calculated. It is a quite dynamic technique, since it can update at runtime and can take into account new data easily via the implementation in ProM.

Bolt, Sepúlveda (2014) – Process remaining time prediction using query catalogs

This article recognizes the previous articles from literature, but indicates that flexibility is missing; therefore the authors introduce query catalogs. A query catalog is according to the article: “A group of non-equivalent partial trace trails from all traces that have occurred in an event log and additional information about each partial trace tail”. Each partial trail is added to the database and can be queried to retrieve the cycle time they needed. The average time until completion is now calculated based on the average time in the catalog where the trace is in. This method mainly focuses on routing and prediction based on partial traces, where less advanced options are derived for calculations on throughput time. Statistically it does not take any distributions or confidence intervals into account, but due to the catalog process it is very dynamic and can handle new cases very well.

Folino, Fuarascio, Pontieri (2013) – Context-Aware Predictions on Business Processes

In order to predict the cycle time for an order, a process performance model is needed according to this article. These models are a form of predictable clustering models and provide estimates of the remaining cycle time. This method especially looks at the context of a trace in order to provide detailed predictions on their cycle time. The metrics for time prediction are made using the average and their standard deviation. This method focuses on routing and there is less focus on throughput time. Statistically it provides a more general approach using only averages and standard deviations to come to a predicted time. It is very dynamic, since it can scale well to new cases.

Di Francescomarion et al. (2015) – Clustering Based Predictive Process Monitoring

This article uses a classification method incorporating decision trees in order to do process predictive monitoring. This monitoring is done in real time tracking traces and comparing them against similar executions of the process based on classification of the trace. Based on a question to the predictor a certain probability of an event going to occur is provided. This is all based on how previous traces
were routed so if certain steps have taken place within the process. The method is mainly based on averages within the clusters where a trace is classified into; therefore it is more upon future routing than on prediction of throughput time. In terms of stochastics also only averages are calculated which makes the model quite deterministic. In terms of dynamism this model really well adapts to new cases, but is harder to adjust after the model is well trained with the data available.

*Pravilovic, Appice, Malerba (2014) – Process mining to forecast the future of running cases*

Based on the past process steps this article aims to predict the future process steps to be taken using a decision tree model to decide what the next activity is. Next to this clustering and classification of what the next step is also a sliding window technique is implemented to only take into account the more recent cases, so that not all previous data is taken into account. The model is first trained offline and then is used online to predict what the next activity of a case is. This model only focuses on routing and not on throughput time. Stochastically it does not look at any characteristics of the throughput time next to the average time, which is only a byproduct here of the algorithm. The technique is also not very dynamic since it is first trained offline and only used online for classification and not for training any more.

<table>
<thead>
<tr>
<th></th>
<th>Routing vs Throughput</th>
<th>Statistical vs Deterministic</th>
<th>Static vs Dynamic</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Routing</td>
<td>Throughput</td>
<td>Deterministic</td>
<td>Statistical to Deterministic</td>
</tr>
<tr>
<td>Sha (2004)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Chen (2010)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Chen (2012)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Chang (2006)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tirkel (2013)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Backus (2006)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Meidan (2011)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Xie (2013)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Ha (2006)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>V. Dongen (2008)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bevacqua (2014)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>V.d. Aalst (2010)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Rogge (2013)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>V.d. Aalst (2011)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Bolt (2014)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Folino (2013)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Di Franc. (2015)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Pravilovic (2014)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1 – Overview of the related work analysis
2.3 Techniques selected

Based on all the related work discussed in this chapter a choice has been made to investigate two different techniques that each cover a different perspective of the field of routing and throughput. Both selected techniques use some form of statistics, are dynamic and also both have an implementation.

The first technique is developed by Rogge-Solti & Weske (2013), which takes stochastics well into account and is very focused on throughput time. The downside of their technique is that after the derivation of the distribution for a certain activity there is no method to calculate the distribution for multiple process steps together. Instead the average value needs to be taken to obtain an estimate to the throughput time until a certain process step. Therefore this method cannot be classified as completely statistical, but as statistical to deterministic.

The second technique is developed by Van der Aalst, Schonenberg & Song (2011) which is mainly based on routing and less on throughput time. It has no statistical distributions in it, but takes process routing better into account, therefore it can be an interesting comparison between the two techniques and also between the new technique developed and the existing ones.

In Table 1 the results from the literature review are presented. As can be seen there are no papers which focus upon statistical methods which indicates a possible gap in the literature. There are also several papers which are both focused on routing and throughput time, but there might not be an implementation available. This is important since the performance of the methods is compared and an implementation is needed to obtain the performance of a method.

In the next chapter the selected techniques are discussed in detail, based upon the gap in the literature and the downsides of both of the selected techniques a new technique is developed. This technique is aimed at using statistical tools to dynamically predict the throughput time while also accounting for possible routings through the process.
3 Techniques for Throughput Time Prediction

In this chapter the selected techniques from the previous chapter is discussed in-depth and also the limitations of these techniques are described. Based on the limitations and findings of the related work review a new technique called Historical Data based Throughput Time Prediction (HD-TTP) is proposed to estimate throughput times and time windows based upon historical data from previous executions. The historical data is stored in a process log; an example process log is presented in Table 2.

In a process log all events executed within the process are logged. First the case id is the key for identifying which process events belong to the same case. For each event the completion time is stored when an event is finished. All techniques discussed in this chapter require a process log to generate estimates for throughput time.

<table>
<thead>
<tr>
<th>Case ID</th>
<th>Event</th>
<th>Completion Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>445</td>
<td>A</td>
<td>08-10-16 22:23</td>
</tr>
<tr>
<td>445</td>
<td>B</td>
<td>08-10-16 22:42</td>
</tr>
<tr>
<td>445</td>
<td>C</td>
<td>08-10-16 22:44</td>
</tr>
<tr>
<td>501</td>
<td>A</td>
<td>12-10-16 20:31</td>
</tr>
<tr>
<td>501</td>
<td>C</td>
<td>12-10-16 20:37</td>
</tr>
<tr>
<td>501</td>
<td>B</td>
<td>12-10-16 20:57</td>
</tr>
<tr>
<td>635</td>
<td>A</td>
<td>22-10-16 03:13</td>
</tr>
<tr>
<td>635</td>
<td>B</td>
<td>22-10-16 03:33</td>
</tr>
<tr>
<td>635</td>
<td>C</td>
<td>22-10-16 03:36</td>
</tr>
</tbody>
</table>

Table 2 – Example process log of a process

3.1 FSM miner & analyser

The technique from Van der Aalst, Schonenberg& Song (2011) needs a process log in order to predict the expected time needed until completion based on process mining methods. In order to generate a prediction for the expected throughput time for an event the following steps are taken:

1. *Mine a transition system using the FSM miner*
2. *Analyze the transition system using the data from the log*
3. *Determine the expected time durations per event*

The first step is to mine a transition system using the FSM miner, the FSM miner is a method implemented in the software package ProM 5.2 to mine a transition system from a process log. According to Van der Aalst et al. (2011) their technique uses the following method: A transition system is a triplet \((S, E, T)\) where \(S\) is the state space, \(E\) is the set of event labels and \(T \subseteq S \times E \times S\) is the transition relation describing the move from one state to another. In order to generate a transition system traces are used, a trace is a finite sequence of events. A transition system starts at an initial state and ends at one of the final states of the system. The possible behavior of the transition system is determined by the routings over the transition system. Traces provide the data for the behavior of the transition system. The mining of a transition system can yield three types of abstractions. The first form of abstraction is the horizon of the prefix used to determine the current state of the system. The second form is filtering for certain events, so only the selected events are taken into account to obtain the current state. These two types of abstraction are based upon the sequence which is taken into account. The third type of abstraction removes the order or frequency of events within a trace, a choice can be made to abstract from a sequence to a multi-set of events or a set of events. The first option to abstract to a multi-set removes the order of events but retains the frequency of their
execution within the trace. The second option to abstract to a set of events also removes the frequency of the execution in a trace and only checks for the mere presence of an event in a trace.

For the mining of the transition system using the FSM miner there is abstracted only to multi-sets of events. A multi-set is a set where next to all different items in the set also the frequency of the item in the set is recorded. An example of a transition system based upon multi-sets is shown in Figure 3.

Second the mined transition system is analyzed using the data from the process log. Using the remaining time for each state of the transition system there is determined how much of the total throughput time is already passed at a certain state. First based upon historical data for the prefix of events the possible start and end time of each trace is determined. A prefix is a sequence of events which is already executed at a specific point in a trace. The remaining time of each prefix is used to calculate how much time the next event in the trace takes place. Using this method a calculation is made for the remaining time of the postfix events. The postfix is the part of the trace of events after the specific point in the trace. In order to obtain estimates for the remaining time duration of a postfix the sample mean of the historical data is calculated. This method is implemented in ProM 5.2 as the FSM Analyzer. It uses the transition system derived by the FSM Miner to obtain remaining throughput time predictions for new cases to be executed.

Third the time expected time duration of each activity needs to be established. The mined and analyzed transition system now yields the elapsed, remaining and sojourn time for each activity. Sojourn time is defined as the average time spent in a particular state of the transition system (Van der Aalst, Schonenber, & Song, 2011). In order to obtain the estimated throughput time of a trace and all events within the trace the trace is analyzed using the analyzed transition system. First the correct state is looked up by using the prefix of the event. The sojourn time at the correct state in the transition system is used to determine the duration of the event. Confidence intervals can be estimated using the standard deviation of the sample based upon the law of large numbers. Therefore it is assumed that all samples are independent and that all measurements are sampled from the same distribution.

3.2 Stochastic petri-nets
The technique from Rogge-Solti & Weske (2013) requires next to a process log also a petri-net of the process to predict throughput times and thereby predict the time needed until completion. The technique is based upon the fact that during the execution of a trace the distribution of the duration of a process step can vary accordingly. So for example if A takes on average 15 time units, then after 10
time units are already processed the remaining time is differently distributed than in the beginning of step A. In order to calculate the expected remaining throughput time the following steps are taken:

1. Generate petri-net process model using mining algorithm
2. Transform petri-net into a stochastic petri-net
3. Determine expected time durations per event

First a process model needs to be generated using a selected mining algorithm. This process model must be in the form of a petri-net in order for the technique from Rogge-Solti & Weske (2013) to be able to handle it as its implementation in ProM 6.5.1. In order to retrieve a process model there can be made use of multiple techniques, for example the alpha algorithm or an inductive miner. The main advantage of the alpha algorithm is that it certainly provides a process model where the traces are repayable. The major problem with this technique is that the implementation in ProM splits all events into a start and an end event, which are both activities without time duration. Therefore an inductive miner is used to obtain a process model with all the activities for the process log provided.

Second the mined petri-net needs to be transformed into a stochastic petri-net. This is done by enriching the petri-net with the data from the process log. For each activity the historical data is used to simulate the process. In order to derive a distribution a sample is derived from the simulated historical data and the best fitting distribution is accepted as the true distribution of the throughput time for that event.

Third the expected time duration of an event need to be extracted. In order to do so the expected value of the distribution for the event needs to be extracted. Using the implementation in ProM 6.5.1 of the technique from Rogge-Solti & Weske (2013) in combination with the stochastic view of the stochastic petri-net the expected value for the distribution can be derived and stored for the event.

The two techniques described have a couple of limitations for while deriving the duration of an event or estimating time windows. The technique by Van der Aalst et al (2011) assumes that the law of large numbers hold and therefore a normal distribution can be used to determine confidence intervals. For the law of large numbers to hold every observation should be from the same distribution, in practice this mainly does not hold. Therefore only the sample means derived by the technique can be used to estimate the remaining cycle time and sojourn time of an activity. The technique of Rogge-Solti & Weske (2013) does not provide any confidence intervals or time windows for events.

The downside of both techniques is that they both do not provide any confidence intervals, since the assumption in the technique by Van der Aalst et al (2011) is violated. Also the technique by Rogge-Solti & Weske (2013) fits a distribution instead of using the empirical distribution. In order to cover this gap in the research on event duration estimation and confidence interval generation for event duration a new technique is presented. This new technique covers both the estimation of event durations using empirical distributions and the generation of confidence intervals for the event duration.
3.3 Historical data based throughput time prediction (HD-TTP)

In order to overcome the lack of statistical techniques and ability to predict time windows of the previous techniques a new technique is developed called Historical Data based Throughput Time Prediction or abbreviated HD-TTP. This technique will use statistical methods to add distributions of the different throughput times together and derive time windows for these processes. A time window is the equivalent of a confidence interval with a certain precision and error. The data used for this method is the historical data obtained from previous process executions.

In this section the way in which the HD-TTP technique works is described. In order for the HD-TTP technique to work the process log with the historical data is needed along with the parameters for the precision and error for the time windows. Next to that a list of events for the reviewed trace is retrieved. The HD-TTP technique will execute the following steps to obtain statistical estimates:

1. **Generate an empirical distribution and segment for each event**
2. **Add together two empirical distributions to a combined distribution**
3. **Calculate the average, minimum window time and maximum window time**
4. **Re-segment the combined empirical distribution**
5. **Repeat the process for the whole list of events within the process**

First for each event in the reviewed trace an empirical distribution is generated based upon the historical data from the process log. For each of the \( n \) previous executions of the event there is a \( 1/n \) probability appointed. By taking the sum of the probabilities from the previous executions of the event a cumulative distribution can be derived as can be seen in Figure 4 for an example event A:

![Distribution of Throughput Time - A](image)

*Figure 4 – Cumulative empirical distribution for event A*
In order to add two of these empirical distributions together each distribution is separated into segments of equal size. For this example with events A and B there is chosen to separate them in ten equal segments. These segments have a starting value and an end value, for example look at event B:

![Distribution of Throughput Time - B](image)

Figure 5 – Cumulative empirical distribution for event B (break x-axis at 40 to 80)

In Figure 5 for example the first segment starts at the cumulative probability 0 and ends at the cumulative probability of 0.1, resulting in the start of the segment at 0 minutes and it ends at 19 minutes. Since the segments are all evenly wide, the probability in each segment is equal. Each segment represents the probability which it covers in the cumulative distribution. In order to add the two empirical distributions together each segment is multiplied with all the other segments of the other distribution, assuming that the two distributions are independent. By each multiplication of two segments a new segment is created by treating both segments as a discrete uniform distribution. Each integer value in the segment has an equal probability within the discrete uniform distribution. So each empirical distribution ηₓ for event X consists of a n number of equally sized segments ϕₙₓ , which each have a minimum and maximum value and a probability. In order to multiply all segments from both distributions and create an empirical distribution for both events combined the following method is used:

For each of the n segments of ηₐ a multiplication is performed with all the m segments from ηₔ. For ϕₙₐ to be multiplied by ϕₘₐ both segments are interpreted as discrete uniform distributions. Therefore all integers from ϕₙₐₗ to ϕₙₐₘₚ have an equal probability and each integer is added to each integer from ϕₘₐ which are all the integers between ϕₙₐₗ to ϕₙₐₘₚ. The probabilities of the added values are calculated by multiplying the probability of the value in ϕₙₐ with the probability of the value in ϕₘₐ. The combined value with its probability is written to a table. If the
value already is in the table the probability is updated with the addition of the calculated probability, if the value is not yet in the table the value is added with the calculated probability.

After all the segments of both distributions are multiplied with each other the table used to store the values and probabilities generated by the multiplications now represent the distribution of both activities. For the events A and B together this results in the following distribution:

![Combined Distribution of Events A + B](image)

Figure 6 – Combined distribution for the events A and B

From Figure 6 the combined empirical distribution for the events A and B. In order to obtain the expected time for the combined execution of both events the value at the cumulative probability of 0.5 is derived, for this example the expected time is 25 minutes. Next to the expected time for the completion of both activities a confidence interval or so called time window can be constructed. A time window has a certain precision and error bounds on the lower an upper part of the time window. The lower error bound marks the earliness value for the time window; this is the error that an event is finished earlier than the time window. The upper error bound marks the lateness value for the time window; this is the error that an event is finished later than the time window. For example a time window with an earliness error of 0.15 and a lateness error of 0.20 gives a time window with a precision of 0.65 (1.00 – 0.15 – 0.20). In order to obtain such a window from Figure 6 the lower bound of the window is found by taking the value at 0.15, the upper bound is found by taking the value at 0.80 (0.15 + 0.65), this leads to a time window starting at 20 minutes and ending at 41 minutes.

After the combined distribution is derived and the averages and time windows are determined another event can be added to the empirical distribution. Therefore the new segments for the combined distribution are determined as before, where each 0.1 probability on the y-axis is another segment.
4 Evaluation of the Techniques on Generated Data
To evaluate the two techniques from literature and the new technique an example process of the receiving, sending and billing of an order is used. The process is used to evaluate the predictive values for each of the techniques. The process consists of three process steps where first an order needs to be received, then either first the order is send to the customer and then the bill is send digitally to the customer or vice versa:

![Example process for evaluation of the techniques](image)

In order to evaluate the different techniques against each other an example dataset is generated for this example process. The properties of the process steps are shown in Table 3 below:

<table>
<thead>
<tr>
<th>Event</th>
<th>Process Time Distribution</th>
<th>Executions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Receive Order</td>
<td>Exponential (15)</td>
<td>1.000</td>
</tr>
<tr>
<td>Send Order</td>
<td>LogNormal (20,30)</td>
<td>1.000</td>
</tr>
<tr>
<td>Send Bill</td>
<td>Exponential (10)</td>
<td>1.000</td>
</tr>
</tbody>
</table>

Table 3 – Process activity overview

Using the BIMP simulator (2016) this process was executed a thousand times and the data was stored on the process duration. There is chosen to investigate the execution order Receive Order, Send Order, Send Bill. For these processes the estimates for the completion time for each event is compared for all the different techniques selected and the new technique developed.

Based upon the historical data the difference between the actual completion times and the predicted completion times by the different techniques can be calculated. In Figure 8 for each technique the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are presented. The MAE is the mean absolute error, where \( \hat{y}_i \) is the predicted value by the technique and \( y_i \) the actual value in the data. The difference between the predicted value and the actual value of the data is calculated for all the
\( n \) number of observations in the data. The MAE takes the average absolute value of the error to compute the MAE; this is calculated in the following way:

\[
MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} |\hat{y}_i - y_i|
\]

Next to the MAE the RMSE is used, the \( \hat{y}_i \) is the predicted value and \( y_i \) the actual value in the data. The RMSE takes the sum of the squared errors and divides that value by the number of observations \( n \). This measure uses the root of the mean squared error to obtain a measure of the precision for the technique. The RMSE is calculated in the following way:

\[
RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}}
\]

Based upon both metrics the following results are obtained for all three techniques and the naïve method using multiple executions of the process of the example dataset. The naïve method uses the average value of the data as predictor for the throughput time of an event.

![Figure 8 – Absolute average difference per activity in arrival time per technique](image)

From Figure 8 can be observed that for the RMSE and the MAE the HD-TTP technique outperforms the technique from Rogge-Solti & Weske (2013), this is likely caused by the different distributions which are hard to handle by the Rogge-Solti & Weske (2013) technique. For the technique from Van der Aalst et al (2011) the high values of the MAE and RMSE is likely caused by the routing from the process which enlarges the variance in the estimated throughput times. The HD-TTP technique handles both these aspects better than the two techniques from literature and the naïve technique.
In order to compare also the time windows derived by the technique the naïve technique is used to derive time windows. The naïve technique uses the average as the centre of the time window and an equal width to each side for the time window. Here is chosen to take a time window the size of 30 percent of the throughput time, which equals to 13.5 minutes. Therefore the time window obtained is the average throughput time until that process step plus or minus 6.75 minutes. This results in the following performance of the time window for the selected cases:

![Performance Naive Window](image)

**Figure 9 – Performance of the naïve window**

The performance results of the naïve window are used as input parameters for the generation of time windows in the HD-TTP technique to use for the time window estimation. The HD-TTP technique performs better than the naïve technique when windows are smaller than 13.5 minutes. The performance of these time windows is equal to the performance of the time windows from the naïve technique. The obtained windows using the HD-TTP technique are shown in Figure 10.

As can be seen the time windows obtained are all smaller than the naïve time windows. About one third of the time windows are less than 50 percent smaller than the naïve window, the other time windows are smaller than 50 percent of the naïve time window. For the two thirds of time windows smaller than 6.75 minutes half of them are even three times smaller than the naïve time windows.
Based upon these measures for the expected throughput time prediction and the window size for the example dataset the HD-TTP technique outperforms the other techniques from literature and the naïve technique on both fronts. In the next chapter a case study is performed to show the performance of the HD-TTP technique on a real-life dataset from Van Opzeeland.
5 Evaluation of the Techniques on Real Data

In the previous chapters the new technique is presented and evaluated on generated data. In this section the evaluation of the techniques on real data is presented using a case study at Van Opzeeland.

5.1 Process description

In this paragraph the process of distribution of cargo at Van Opzeeland is described. The company characteristics and the process characteristics are discussed and explained. After the process is described the link with the data from Van Opzeeland is described.

Van Opzeeland is a transportation company with about 150 employees, 65 trucks and 35,000 m² of warehouse space. Its focus is on retail distribution, specifically targeting the distribution of shoes, clothing, pet supplies, high-tech and cosmetics. However, several offices and petrol stations are also supplied by Van Opzeeland. All these industries combined result in about five thousand delivery points every week. Next to these delivery routes Van Opzeeland also performs warehousing, but for this report it will not be included within the scope of the process.

Before each truck can leave the yard, it needs to be loaded at the warehouse with the goods it will distribute at the trip driven that day. Each truck leaves the yard of the warehouse and drives to the first order of the trip that day. A trip is a process instance of the standard route a truck takes. A standard route is the main process a truck follows. These routes consist out of standard set of orders for that a trip. Each trip can consist out of the standard orders with some adjustments to it. An order is a location where certain goods need to be delivered; the unloading at the location is called a stop, whereas the process of driving between location A and B is called a leg. Therefore, each trip is a collection of legs and stops which are executed in a predefined order. After all the legs and all stops are executed a truck will return to the warehouse. This process is shown in Figure 11.

![Figure 11 – Process model Van Opzeeland in BPMN](image)

From this process model it looks like a straight forward process, but in practice this process is very heavily relying on data, since driving to the next location differs per order in a route and also the unloading time is dependent on the location where the goods are delivered. The decision to finish the route is based on the fact whether a trip still contains orders which need to be delivered. For example...
the route with number 1484000 is shown in Figure 12 where each blue line between two dots is a leg and each address dot is an unloading and loading location. Identical trips are seldom executed, but often variations of this route are executed, which creates the necessity for data aggregation.

![Figure 12 – Example route number 1484000 in the center of Rotterdam](image)

The data for this case study needs to be aggregated; the choice has been made to calculate transportation legs between postal code areas instead of between single addresses, because the amount of times the same leg between two exact locations is executed is very low, while the same areas are visited multiple times. After a transportation leg is completed the cargo of the truck should be unloaded from the truck to the warehouse of the customer and return packages should be put in the truck. These unloading times are needed at every instance of a trip and can be very short in case that there is only a small amount of cargo to transfer or rather large when a lot of cargo needs to be restacked from a cart. These unloading locations are also quite sparse, therefore within data aggregation a choice has been made to aggregate to general categories of types of unloading locations.
5.2 Dataset overview

In this section the data used for the comparison of the techniques for the Van Opzeeland is discussed. Next to the characteristics of the data also the context is addressed. After describing the data, the data is cleaned and prepared in order to use it in the technique.

5.2.1 Data properties

The dataset obtained from Van Opzeeland contains all the trips the trucks of Van Opzeeland have conducted in the period of 01 January 2016 to 22 April 2016. This dataset contains 6558 trips and each trip consists on average of 19 orders per trip, accumulating to 126,372 orders. In the next paragraphs aspects of the data are discussed.

For each order it is recorded to whom and where the order is delivered. Per order the clients’ complete address, including postal code, street name and number and city are known. These addresses are also coded in the data using a unique address code for each address. This data can be used in order to retrieve the exact locations where the orders are delivered.

Next to the location where the order needs to be delivered also the information about what needs to be delivered is available, for example the number of pallets or containers, the number of meters of load floor used for that order. Next to the cargo size there is recorded whether it is a distribution order or a return order, as well as the type of vehicle and driver. This data makes it possible to exactly retrieve what cargo is delivered at the location.

Regarding dates and timestamps in the data there are several different types in the dataset. The most important times are the actual arrival and departure times for orders and routes. Next to these the expected arrival time, can be used to compare the time windows from Van Opzeeland. Next to the timestamps and accompanying dates, data is available on the amount of rest time a driver has taken at a certain route as well as the total time for all stops and driving time per route. In Appendix A an overview of the columns of both tables is provided.

5.2.2 Data cleaning

In order to obtain correct and clear results from the analysis data errors in the data need to be removed from the data. This means that coding errors and missing values need to be corrected and duplicates of the same order need to be resolved within the dataset. In order to do so the following data cleaning procedures have been applied. A choice has been made to delete missing values, since imputing the missing values would lead to larger uncertainty in the data and therefore to less reliable results.

An order within the transportation schedule has a starting time for unloading the vehicle and an end time for the unloading. This defines when a truck has arrived at certain locations, which are the start time and also the departure time at a certain location which is captured by the end time. Without these times it is impossible to reconstruct the route of a truck and the times needed for driving and unloading. If an order does not have either one of them, it can be assumed that this order is skipped in
the trip and therefore can be left out of the dataset. By executing this cleaning operation, 2,262 orders are deleted from the dataset based on the absence of both start and end times.

For orders for which either the start time or the end time is missing from the order it is more difficult to interpret the trip, for instance when did the truck depart from that location or arrive at that location. Furthermore, since this data is missing it is unclear how the traveling time and unload time for the other orders of the trip should be determined. Due to this uncertainty of missing values, which might affect other times of the trip a choice has been made to delete the whole trip from the dataset. The result of this cleaning operation was to delete 334 trips, containing together 6,750 orders, from the dataset.

In the transportation sector it is possible that at the same location different orders are delivered. This means that several packages are delivered to the same address after each other, which in practice holds that multiple packages are delivered in one go. In the database these orders are typically logged after each other, as all the orders together yield the total unload time for that address. In order to prepare the data in such a way that each leg is between two different locations these orders are merged. This is based on the address code in the data which supplies every address with a unique code. For these cases the earliest start time, the latest end time and the sum of the unloading times are taken into account. Also the amount of goods to be unloaded is summed up together. By merging all these orders together for which the address is equal to the previous one, a total number of 30,664 orders have been merged together.

After cleaning the data, 88,381 orders are remaining in the dataset which are distributed within 6,224 trips. In the next section the data is prepared for further analysis.

5.2.3 Data preparation
Next to cleaning the data from obsolete orders or missing data the data will also be prepared using some calculations to extract more details about each order. For this dataset the traveling time between two orders, defined as the duration of a leg, is determined and added to the dataset. Furthermore for aggregation purposes the postal code is used to generate a location identifier, upon which is used to aggregate the results and also the unloading times is grouped per category of type of business trucks is unloading to.

The travel time per order is calculated because it is not by default available in the data. The travel time is the time passed between the departure at the last order and the arrival at the current one. In order to calculate the travel time it should be determined first whether the leg is the first of a route or not. The first leg departs from the warehouse and has the start time of the route as departure time and the arrival time is the start time of the first order. The travel time is calculated by deducting the start time from the arrival time of the first order. If an order is not the first of a route it is determined by deducting the end time of the previous order from the start time of the current one. Each travel time is stored with the corresponding order in the database.
Some routes have a missing start time, therefore it is impossible to calculate the travel time for the first order of that route. Therefore these orders should be deleted from the dataset; by doing so 779 orders are deleted from the dataset. Imputation is not an option since this data is not recoverable from other data items in the dataset. Furthermore, orders with a negative travel time are also deleted from the dataset. A negative travel time can occur due to a later departure time recorded at the warehouse for a route than the actual arrival of the truck at the first customer. This has resulted in the deletion of 111 orders.

Within the transportation network identical legs are not often reoccurring, even since Van Opzeeland uses several usual routes, but due to planning algorithms sometimes one order is before or after another. Therefore most identical legs are only driven once or twice, which increases the difficulty of deriving statistics. In order to do so, aggregation of the data is needed. Instead of using the address code to code a leg from location A to location B a new postal code key is generated. This is a code consisting of the country letter where the address is residing and for the Netherlands the first four numbers of the postal code are included in the key. This groups addresses within the same neighbourhood together. For other nations the whole address code is incorporated since it is not guaranteed how locations are divided within those postal code regions. Due to missing the postal code, two orders are deleted from the dataset.

Based on the client code groups of the same type of stores are identified in the dataset. Next to the description provided in the dataset a more general description for groups of client codes is generated based on the type of businesses, according to their description. Seven categories can be identified, for which a client type can be in, these are: Hospitality, Animals, Shoes, Retail, Drugstore, Expedition and Other. These categories are indicated by the following client codes.

<table>
<thead>
<tr>
<th>Category</th>
<th>Client Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>FEC, FP, JPT, JA, NER, RC</td>
</tr>
<tr>
<td>Drugstore</td>
<td>BM, GEF, HR, IPXL, IPXLM, KR</td>
</tr>
<tr>
<td>Expedition</td>
<td>BRTC, ESSERS, JLN, KUI, KV, TPC, ZAN</td>
</tr>
<tr>
<td>Hospitality</td>
<td>HS, KIV, KKS, LFE, NABT, RAD, SCA, SWG, VERB, ZUID</td>
</tr>
<tr>
<td>Retail</td>
<td>AL, AMA, BEV, CB, FC, HM, HOG, KI, LAR, MLO, NABR, OPZEE, PAT, POR, RIJ</td>
</tr>
<tr>
<td>Shoes</td>
<td>FL, FLA, HRG, NEL, SA, SS, TAK, VH, YARI</td>
</tr>
<tr>
<td>Other</td>
<td>ALA, ALR, BAXL, BE, BER, FF</td>
</tr>
</tbody>
</table>

Table 4 – Categorized client codes for Van Opzeeland
5.3 Data properties

After the data cleaning and preparation some descriptive about the data can be derived. In this section the travel times and unloading times specifically are reviewed. The data properties are reviewed in order to provide insight into the data used in this case study. Furthermore there is aggregation applied to make the data more usable for the new technique.

5.3.1 Legs and travel times

The first step in the process is to travel to the delivery location. From the dataset several descriptive insights are extracted in order to provide a detailed overview of the structure of this data. First the frequency of all legs is taken into account; this yields how many times the same leg is executed between two locations. For the unique addresses combined to legs this results in the distribution as in Figure 13, one can see from the distribution that the maximum number of times a leg is executed is only 78 with about 88 thousand orders; this is quite a low amount of time the same trip is executed.

![Frequencies of Legs](image)

**Figure 13 – Frequencies of transportation legs**

Therefore, instead of the single addresses the postal code areas is used. This leads to the following frequencies of leg executions as in Figure 14. One can see here that still there are several legs which only occur once or a couple of times, but many legs are executed multiple times, which is a logical result from the aggregation. On the horizontal axis the number of times a leg is executed is plotted whereas on the vertical axis the amount of times it is the case that a leg is executed that many times is plotted. Note however that this is a logarithmic scale. If more data is available it can be assumed that this distribution of the frequencies will shift more to the right since there is then more information about the process execution in the past.
Figure 14 – Aggregated frequencies of transportation legs

These legs all have their individual travel time per occurrence, when plotted all the travel times and their frequencies are presented in Figure 15. Next to the travel times presented in the graph there are still 43 travel times which have only a single occurrence and are larger than 210 minutes and therefore do not show up in a graph with a logarithmic scale. Therefore the graph size has been reduces to 210 minutes on the horizontal axis but for the analysis these other 43 travel times are also taken into account for further analysis.

Figure 15 – Frequency of travel times
It can be seen that there are a lot of trips with only zero minutes traveling time, this can be caused by the fact that orders which are delivered in the same street take no time to travel to since the truck does not move. What is typical for these distribution routes is that we see a lot of small traveling times, which indicates that the addresses are quite close to each other. The average traveling time is 16.42 minutes with a standard deviation of 19.85 minutes.

5.3.2 Unloading times
After arriving at the location the truck is unloaded and possibly loaded with a return freight. This process is executed for every order and takes a certain amount of time. Unloading times are different for each address, but are assumed to rely on the type of store and the amount of goods needed to be un-loaded. In order to compare this first an overview of the overall unloading times are provided, but also a division on the type of customer, provided by the client code. First the non-aggregated frequencies per location are provided in Figure 16.

![Frequencies of Stops](image)

**Figure 16 – Frequencies of Stops for locations**

When all unique locations, 15,766 in total, are considered for the frequencies of a stop the maximum frequency is only 145. On the horizontal axis the number of times a stop is executed is given, whereas the frequency for each of these in the dataset is given on the vertical axis. Here it can be seen that there are a lot of locations which are only visited a couple of times during the period whereas some others more often. If more data is available this graph will probably shift more to the right due to that the address base of Van Opzeeland does not shift as fast.
The duration of a stop is, as can be seen from Figure 17, normally not too long. This is logical since most routes within the dataset are distribution rounds for which only parts of the cargo need to be unloaded. The average duration of a stop is 13.16 minutes with a standard deviation of 16.34 minutes.

![Duration of Stops](image)

**Figure 17 – Duration of Stops in minutes**

Different clients yield also different stop times, therefore also the split on client code and their categories are provided. Therefore the average and standard deviations for each client code are provided in Appendix B and for the categories these are provided in Table 5. Using a two-sided unpaired t-test with an alpha value of 0.05 only the categories of Expedition and Hospitality are not significantly different from each other. But due to the differences on opening hours and types of cargo transported the decision is made to keep both categories in the data.

<table>
<thead>
<tr>
<th>Category</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animals</td>
<td>9.46</td>
<td>14.50</td>
</tr>
<tr>
<td>Drugstore</td>
<td>15.86</td>
<td>15.31</td>
</tr>
<tr>
<td>Expedition</td>
<td>12.98</td>
<td>13.67</td>
</tr>
<tr>
<td>Hospitality</td>
<td>13.24</td>
<td>14.43</td>
</tr>
<tr>
<td>Other</td>
<td>19.64</td>
<td>30.42</td>
</tr>
<tr>
<td>Retail</td>
<td>13.62</td>
<td>13.62</td>
</tr>
<tr>
<td>Shoes</td>
<td>16.82</td>
<td>19.89</td>
</tr>
</tbody>
</table>

**Table 5 - Overview of Categories with their average and standard deviation stop time in minutes**

In addition to this tabular data, also a graphical representation is made of the different distributions of the stop times per category. In Figure 18 the different distributions are shown. One can see that all the different stop times per category are from a different distribution than another. They all have about the same shape, but the height and width of the curves are all different.
From Figure 19 one can see that the frequency of the number of stops per client code greatly increased comparing to the frequency of stops per location. This difference is logically explained by the aggregation applied by the client code, which combines different locations under the same client code. Based on the increased frequency more accurate measures can be taken to estimate the stop time for any truck stopping at a location with that client code, provided that clients within the same client code have the same properties with regards of cargo and therefore unloading times are also comparable.
From Table 5 one can see that for all the main categories the average stop time lies between about 13 and 17 minutes. Only the animal branch is quicker than all other categories with only an average stop time just less than 10 minutes. The category of the other client codes has the longest stops with just fewer than 20 minutes on average.

The standard deviation on all categories is quite large, for all categories even larger than their average stop time. This indicates that the variability in the unloading times are quite large, this can be caused by the fact that some client codes are visited less often and therefore only a few data points are available which could enhance the variability of the data. Some client codes have a standard deviation, as can be seen in Appendix B, of zero. It is not the case that these client codes are very precise, but there is only one data point available for those client codes, which makes the standard deviation zero. From Figure 20 it can be seen that when aggregating to one level higher there are at 3,000 stops per category, therefore aggregation can provide more reliable statistics. For the calculations the categories of stops is used to estimate the stop times.

5.3.3 Time windows
Within the transportation sector time windows are provided to customers to indicate the estimated window of arrival at the customer. This window size is the difference between the beginning of a window and the end of a window. The error of a time window can be split into two parts, earliness and lateness. Logically, if the window is increased, the error will decrease. Clients however want tight time windows instead of windows with the span of a whole day; therefore first the error of the estimated arrival time in comparison with the real arrival time is reviewed. After that the current window estimates from Van Opzeeland are discussed.

![Frequency of Stops at Category](image-url)
In order to obtain the error within a route, the difference is calculated between the estimated travel time as available in the data and the actual arrival time as can be seen in Figure 21. The average error within this dataset is 1.29 minutes and the standard deviation is 69.92 minutes. This shows the big error around the estimated arrival time and the need for time windows.

![Time Difference Estimated Arrival Time](image)

**Figure 21 – Difference between estimated arrival time and actual arrival time distribution**

Due to this error in the estimated time of arrival time windows are provided to the customers. Currently the time windows set by Van Opzeeland is calculated by taking the estimated arrival time rounded by 15 minutes and adding two hours to create a two hour window. From this strategy it can be determined that being too late is a more severe problem for Van Opzeeland than being too early. In the next paragraphs the difference between one and two hour windows is discussed and also the difference between the windows from Van Opzeeland and a balanced window, as would make sense since the distribution of the error is quite balanced around zero.

Based on the two hour window as estimated by Van Opzeeland it can be seen that in 51 percent of the times the truck will arrive too early at the destination, in 46 percent of the times the truck will arrive within the window and only in 3 percent of the times the truck will arrive too late. As indeed only a small percentage of times orders are too late, it makes sense how Van Opzeeland calculates their windows. If a balanced window was used of two hours, so one hour before the estimated arrival time and one hour after the estimated arrival time it makes a higher on within window percentage. One can then see that only in 11 percent of the times the truck are too early and the times the truck is within the window at the customer is increased from 46 percent to 77 percent. Only the number of times a truck is too late is greatly increased from 3 percent to 12 percent.
When looking at only one hour windows the times a truck is early according to the Van Opzeeland windows do not change, since that window only changes the number of times they are too late. This leads to an increased late rate from 3 percent to 12 percent, an early rate of 51 percent and a decrease of percentage of orders being within the time window to only 37 percent. Once a balanced window is in place it shows that 24 percent of the time a truck is late and also 24 percent of the time the truck is early. In 52 percent of the time the truck is within the window, which is a big difference to the only 37 percent for the windows as proposed by Van Opzeeland.
In order to compare the different techniques with each other, results are obtained based upon the Van Opzeeland dataset. Therefore multiple routes which are executed many times are selected to be used in the comparison of these techniques.

The estimation technique from Van Opzeeland is based on the estimated arrival times retrieved from the PTV route scheduling program. These estimated times are based on a speed profile of a certain truck and the route the truck will drive. A speed profile is a list of types of roads and the average speed a truck can drive on that type of road. For example a truck drives on average 70 km/h on a highway, so if there is 35 km of highway in the route, 30 minutes are added to the travel time. Also for each location
a certain amount of stop time is planned to unload the truck at the location. This stop time is based upon the amount of cargo and weather it needs to be unstacked or not. The window times from Van Opzeeland are calculated based on this estimated arrival time. The window is obtained by rounding the estimated arrival time to the next quarter, which is the lower bound and an addition of two hours to that time forms the upper bound of the window.

All techniques require the process steps from the route which is to be executed. But next to a process model the selected techniques from the related work section also need a process log. In order to derive a process log from the dataset an analysis on similar trips is conducted. For all the trips a similarity score is calculated comparing the current route to the trip. Here the multiset of activities is compared for both trips and the difference is expressed in a percentage. So for example if a route exists of activity A,B,C,D and another trip exists of only A,B this resolves in a similarity score of 0.50. Routes that deviate no more than 0.50 from the route which is to be executed are considered an instance of the same process. The selected routes are put into a process log, which can be used in ProM, where both selected techniques have their implementation.

5.4.1 Techniques from literature

The time per activity for the technique of Van der Aalst (2011) is calculated by taking the difference between the predicted remaining time before the start of the executed activity and the predicted remaining time at the end of the activity. These times were retrieved via the implementation in ProM provided by the authors of the paper. First the FSM miner is used to derive a transition system, afterwards the FSM analyser is used to enrich the transition system with the log data, making use of multisets. The data needed is derived by looking at each step of the selected route in the process and determining the time per activity.

The technique of Van der Aalst et al (2011) derives its own process model using the FSM miner, but the technique of Rogge-Solti & Weske (2013) requires a petri-net for the technique to work. In order to mine a petri-net where each activity is represented by one transition an inductive miner is used. Other techniques for extracting a petri-net, like the alpha algorithm, split the start and the end of an event and therefore cannot be used to generate a petri-net for this technique. Using the petri-net it is enriched with statistical distributions based upon the data from the process log. The average values are denoted for each process step.

In order to obtain the arrival times for each order within a trip the average values obtained from the previous steps need to be transformed from the average duration of the order to the average arrival time of the order. In order to do so the throughput times until the order is reached in the trip are summed up. This results in the time needed from the start of the trip until arrival at each order. The actual arrival time is calculated by adding the time until arrival to the start time of the trip.
5.4.2 Performance of the techniques

Based upon these arrival times at each order the difference between the actual arrival times and the predicted arrival times by the different techniques can be calculated. In Figure 26 for each technique the Mean Absolute Error (MAE) and the Root Mean Square Error (RMSE) are presented. As can be seen the two selected techniques from the related work section perform at just above 30 minutes, where Van Opzeeland is slightly better and has a value for the MAE just less than 30 minutes. For the new technique this value is only at 15 minutes and therefore half of the MAE as the other techniques.

![Figure 26 – Absolute average difference per order in arrival time per technique](image)

Next to the MAE the RMSE is calculated, which results in a high value for the technique by Rogge-Solti & Weske (2013) and the technique used by Van Opzeeland. For the technique by Van der Aalst et al (2011) this value is about the same as the MAE, where for the HD-TTP technique the RMSE value is 60 percent lower than for Van Opzeeland and Rogge-Solti & Weske (2013), and 35 percent lower than for Van der Aalst et al (2011). This shows that for both measures the HD-TTP technique outperforms the existing techniques.
The normal window size provided by Van Opzeeland is two hours. Optimally the window time is reduced to provide a better indication of the arrival time at the client. Due to the stochastic methods the time windows from the HD-TTP technique fluctuate in size. In Figure 27 the different sizes of the windows obtained by the technique are provided, aggregated by their size.

These windows are derived with the exact same earliness and lateness probabilities as the original windows from Van Opzeeland, for which 51 percent earliness is allowed and only 3 percent lateness. Since the precision of the windows is equal the window size should be smaller to indicate that the technique predicts better time windows than the current method from Van Opzeeland. In 70 percent of the cases the window size is smaller than two hours, where 35 percent is even smaller than one hour. Based on this metrics one can conclude that in the majority of cases the HD-TTP technique outperforms the time windows as provided by Van Opzeeland.

Given the comparison of the different techniques, one can conclude that the HD-TTP technique as proposed in the previous chapter out performs all the existing techniques on the calculation of the average arrival time. Based on the comparison to the time windows as derived by Van Opzeeland the new technique performs better in 70 percent of the cases whereas the provided window is smaller. The precision for these windows is equally; this is ensured by the method for window estimation, as also proposed accompanying the new technique.

**Figure 27 – Window sizes obtained using the new technique**
5.5 Limitations of the case study
Within this case study there has been made use of the data from Van Opzeeland. The process behind all this data is quite linear per route and therefore shows only the performance for strictly linear processes.

Next to that an issue with the data is that this is the originally planned data combined with the true execution times. Last minute changes to the schedule are not taken into account in the dataset. Therefore in practice probably the performance of the time windows is a bit better than presented in this thesis, because the adjusted times also provide a new time window which is adjusted for the last minute changes. These last minute changes take place rarely, therefore the influence is minimal.

5.6 Future research on the case study
For future research at Van Opzeeland there can be several opportunities to explore. The HD-TTP technique currently only relies on the previous execution times of an activity and predicts a time window. But constraints like loading windows at the customer, wherein a truck may only unload, or mandatory resting times are not taken into account. These factors could influence the processing time and therefore the true processing time might be different. A richer dataset combined with an extension of the technique might enhance the already promising results. Furthermore new window estimation techniques, especially during the process, can enable smaller time windows for orders later in the trip. Incorporation of these techniques can further enhance the technique and increase precision of time windows for customers from Van Opzeeland, which can yield a competitive advantage for Van Opzeeland.
Routing Extension of the HD-TTP Technique

In the previous case study a mainly linear process is investigated, therefore this case study will focus more on the routing within the process and the different variants of a process which can exist. First the handling of routing within the HD-TTP technique is discussed, and then a comparison is made between the HD-TTP technique, the existing techniques and a naïve method.

Routing is the existence of different executions of the same process. Each of the different executions, or order of events, is a variant on the process. For example if there is a process with the events A, B, C and D and there is a choice between B and C in the process routing exists in the process. The possible variants for this process can be A-B-D or A-C-D. A variant is thus a unique order of events for a process execution.

The technique, as described in chapter 4 of this report, combines empirical distributions of events into a combined empirical distribution. If one does this for the same order of events, also the same empirical distributions are derived. This assumed that the dataset which is used for the generation of the empirical distributions has not changed. The same events in the same order leads to identical time windows and average throughput times predicted by the HD-TTP technique. For a process with different variants the HD-TTP technique does not need to calculate the throughput times for each process instance separately but can compute just the values for each variant of the process. So for example if there are the two variants A-B-D and A-C-D, only once for each variant the empirical distribution has to be calculated. So there are two empirical distributions generated: one for ABD and one for ACD. This saves a lot of computation time in comparison to generate the empirical distributions for each case in a variant. Therefore for analysis of routing in a process the HD-TTP technique only requires the variants in the process:

1. Determine in which variants the process is executed
2. Calculate the time windows for each variant
3. Determine the probability of the variant
4. Calculate the expected time window based upon variant probability

First the different variants in a process need to be extracted, this to retrieve information on how many times a variant is executed and which variants are in place in the process. These variants are numbered and their frequency is denoted for further analysis.

After all variants are known, for each variant the technique is applied. This is done like for any other process where for each event the empirical distribution is derived and added to the empirical distribution of the next event. This method is performed until there is an empirical distribution until the end of the variant, this in order to retrieve a time window for all intermediate steps and for the total throughput time. For the variant the average throughput time and the time windows for each event are stored and are used to calculate the expected time window for the process.
Third the probability of each variant needs to be determined. Some variants of a process occur more often than others. The probability of occurrence of a variant is determined by dividing the frequency of the variant by the total number of variants. This probability indicates what the likelihood is that a variant is executed within a process. This probability is stored accompanied by the variant and the time window associated with the variant.

Last the overall time window needs to be determined for all variants, therefore the probability of each time window is assessed. Based upon the time windows for all the variants with their probability the average time window is extracted. This time window can be used to estimate the duration of an unknown next instance of the process.
7 Evaluation of the Extension on Real Data

In order to validate the technique also for the usage within processes using routing the dataset from the BPI challenge 2015 is retrieved. The dataset has some issues with very long throughput times, some even extending over four years for a process instance. In order to reduce this there is filtered on cases which lasts a maximum of ninety days, about three months. This covers 70 percent of the cases and thereby represents a large part of the dataset. The number of variants in the dataset is large; about 814 variants are possible, where only a couple of variants are executed multiple times.

The maximum frequency for a variant in this dataset is only eight times and about 99 percent of the variants are only executed once. Therefore working with a technique using prefixes is not a viable strategy, since there are so many different variants that the repetition of each prefix is low. Instead of using prefixes the new technique developed is used based upon the historical execution of tasks. 95 percent of the tasks executed are performed more than 20 times. The minimum number of times an activity is executed for the new technique is twenty times, therefore the new technique is chosen to use for generation of time windows for the process step. For process steps executed less than twenty times the average throughput time is taken as the value for the duration. Before processing the throughput time is calculated for each event in days.

A naïve technique is used to calculate the initial time window and average throughput time. For the expected throughput time the average of the data for the current activity is selected. The time window is estimated by taking the average throughput time from the data and adds seven days to the front. This so it is known in which week the event will take place.

![Performance Naïve Window](image)

*Figure 18 – Performance Naïve Time window for all variants selected*
All the variants which have multiple executions within the dataset are used to compare the different techniques to each other. Based upon the historical data from these executions and the naïve method the performance as in Figure 28 is obtained. Here can be seen that in 49 percent of the cases the event was started earlier than the time window, in 45 percent of the cases the event was started in the window and in only 6 percent of the cases the event was started after the time window. Using the values from Figure 28 for the precision, earliness and lateness parameters the HD-TTP technique derived new windows with various sizes as shown in Figure 29.

![Window Size HD-TTP](image)

**Figure 29 – Window size obtained using the HD-TTP technique**

Only in 17 percent of the events the time windows obtained by the technique were smaller than the time windows obtained using the naïve method. In 17 percent of the cases the time windows were less than double the naïve time window and for 66 percent of the time windows the time window was larger than double the naïve time window. The HD-TTP technique performs significantly worse on this dataset than in the previous examples. This might be explained by the very large variance in throughput times in this dataset, sometimes an event can take minutes sometimes it can take weeks. Due to these large variances the empirical distributions also become wider and therefore the time windows become larger.
From Figure 30 can be seen that the HD-TTP technique performs worse on both metrics for the expected throughput time. Again this is probably mainly due to the large variations in throughput times in this dataset. The technique is quite sensitive to these data properties, since the empirical distributions take all values into account.

### 7.1 Limitations of the case study

The method how the HD-TTP technique can be used for processes where routing plays a role is well explained, but the technique cannot out-perform the other techniques due to the large variance in the data. A dataset with less variance or richer data in order to perform more data cleaning could be used in future case studies to analyze the capabilities on routing for the new technique, this mainly to single out the effect of routing and remove the influence of the variance within the process.

### 7.2 Future research on the case study

The case study shows that the HD-TTP technique performs worse than the naïve technique. Therefore improvements on the new technique can be investigated. Also the evaluation of the new technique could be repeated on a different dataset with less variation in the throughput times. The different dataset might isolate the effect of routing in the process and therefore provide better insight into the performance of the HD-TTP technique.

<table>
<thead>
<tr>
<th>Technique</th>
<th>MAE</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HD-TTP</td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td>Naïve</td>
<td>10</td>
<td>15</td>
</tr>
</tbody>
</table>

Figure 30 – Difference estimated throughput times
8 Conclusion

In this chapter the main research question as presented in chapter 1 is answered. Furthermore, the limitations of the study are addressed and future research directions are indicated.

8.1 Research question

The goal of this research was to develop a new technique for processing time and time window prediction. This technique provides the answer to the main research question:

*How can reliable and statistics based estimates for throughput times and time windows be generated?*

The technique was developed by first studying the literature for the existing techniques. The gaps observed in the literature are used as a starting point for the development of the new technique. Especially the gap on the use of statistical methods in the prediction of processing time is solved by the new technique. Also the lack of statistical time window prediction is solved by the new technique.

The technique works by calculating probability distributions for the processing times and waiting times for individual tasks that occur in a process and then making it possible to add up these processing times and waiting times to compute the probability distribution of the start time or end time of a particular task and the process as a whole. Two variants of the technique are presented, one for strictly linear processes (i.e. processes where the path that a customer will take is known at the moment the customer comes in) and one for processes with routing (i.e. processes where there are multiple possible paths that a customer can take, each with its own probability).

The technique is evaluated on an artificially generated dataset. The dataset was generated to study the effects of different arrival time distributions and processing time distributions of individual tasks on the accuracy of the predicted time windows and overall processing time. Next to the artificial dataset two case studies are used to evaluate the techniques.

In order to compare the new technique to the techniques from literature and the naïve technique there has been made comparisons on the predicted processing time using the MAE and the RMSE. Next to this comparison also the estimated time windows are compared to the time windows generated by the naïve technique.

For the artificially generated dataset the new technique outperforms the other techniques on the MAE and the RMSE by about 50 percent. For the time windows also all estimated time windows are smaller than in the naïve technique, with the same precision. For the real-life case study with the strictly linear process at Van Opzeeland the new technique performs better than the other techniques, where in 70 percent of the cases smaller time windows are provided yielding the same precision. Half of the smaller time windows are reduced by at least 50 percent compared to the time windows from Van Opzeeland. Only for the real-life case study for the process with routing the new technique does not
perform better than the naïve technique, also for the time windows only 17 percent of the time windows is smaller than the time windows generated by the naïve technique.

8.2 Limitations
The limitation of the new technique is mainly that the technique does not perform well for data with large variability in the throughput times; also data sets with too few executions for a certain process step cannot be handled by the technique. In order to make the technique even more generalizable the new technique needs to be improved on the limitation of the cases where process steps are executed too few times to construct an empirical distribution with a specified number of segments.

8.3 Future research
For future research adaptive time windows can be researched, now the technique provides static time windows after calculation before a case is executed. During process execution process steps are executed and therefore statistical throughput times now resolve to one value, therefore in the next process steps the total amount of variance is reduced and new calculations could be made. Also the implication of the uniformly distributed segments can be investigated and a non-uniform distributed method might be implemented. The new technique can also be improved by solving the problem for process steps with too few executions.
Bibliography


## Appendix A – Table overview

### Routes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ritnr</td>
<td>INT</td>
<td>The unique identifier for this route</td>
</tr>
<tr>
<td>Srt</td>
<td>STR</td>
<td>The type of route where D is Distribution</td>
</tr>
<tr>
<td>Datum</td>
<td>Date</td>
<td>The date at which the order is executed</td>
</tr>
<tr>
<td>Trucker</td>
<td>STR</td>
<td>Company name who executed the route</td>
</tr>
<tr>
<td>Vrtnr</td>
<td>INT</td>
<td>Number indicating the leaving truck</td>
</tr>
<tr>
<td>Type</td>
<td>STR</td>
<td>Type of vehicle used for delivery</td>
</tr>
<tr>
<td>Route</td>
<td>STR</td>
<td>Textual identifier of route</td>
</tr>
<tr>
<td>Persnr</td>
<td>INT</td>
<td>Personnel number of the driver of the truck</td>
</tr>
<tr>
<td>Naam</td>
<td>STR</td>
<td>Name of the driver of the truck</td>
</tr>
<tr>
<td>Plan duur</td>
<td>INT</td>
<td>Estimated duration of the route</td>
</tr>
<tr>
<td>Ritduur</td>
<td>INT</td>
<td>Actual duration of the route</td>
</tr>
<tr>
<td>Afwritduur</td>
<td>INT</td>
<td>Difference between estimation and actual route duration</td>
</tr>
<tr>
<td>Stop tijd</td>
<td>INT</td>
<td>Time needed for stops on the route</td>
</tr>
<tr>
<td>Stops</td>
<td>INT</td>
<td>Number of stops on the route</td>
</tr>
<tr>
<td>OVE</td>
<td>Float</td>
<td>Number of meters loading floor needed for that route</td>
</tr>
<tr>
<td>Plan km</td>
<td>INT</td>
<td>Estimated number of kilometers needed for the route</td>
</tr>
<tr>
<td>KM</td>
<td>INT</td>
<td>Actual number of kilometers needed for the route</td>
</tr>
<tr>
<td>Afw km</td>
<td>INT</td>
<td>Difference between the estimated and actual number of kilometers</td>
</tr>
<tr>
<td>Start</td>
<td>Time</td>
<td>Start of shift of the driver</td>
</tr>
<tr>
<td>Vertrek</td>
<td>Time</td>
<td>Actual departure of the truck for the route</td>
</tr>
<tr>
<td>Aank</td>
<td>Time</td>
<td>Actual arrival of the truck at the depot</td>
</tr>
<tr>
<td>Einddienst</td>
<td>Time</td>
<td>End of shift of the driver</td>
</tr>
<tr>
<td>Rust</td>
<td>INT</td>
<td>Time needed for resting during the route</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th><strong>Orders</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>Order</td>
</tr>
<tr>
<td>Client</td>
</tr>
<tr>
<td>Filiaal</td>
</tr>
<tr>
<td>Naam</td>
</tr>
<tr>
<td>Adres</td>
</tr>
<tr>
<td>PC</td>
</tr>
<tr>
<td>Plaats</td>
</tr>
<tr>
<td>Datum</td>
</tr>
<tr>
<td>Ritnr</td>
</tr>
<tr>
<td>Volgnr</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Adr code</td>
</tr>
<tr>
<td>Land</td>
</tr>
<tr>
<td>Van</td>
</tr>
<tr>
<td>Tot</td>
</tr>
<tr>
<td>OVE</td>
</tr>
<tr>
<td>Plan ak</td>
</tr>
<tr>
<td>Stop tijd</td>
</tr>
<tr>
<td>Los tijd</td>
</tr>
<tr>
<td>Start tijd</td>
</tr>
<tr>
<td>Eindtijd</td>
</tr>
<tr>
<td>RC</td>
</tr>
<tr>
<td>Pall</td>
</tr>
<tr>
<td>Vrt Type</td>
</tr>
</tbody>
</table>
## Appendix B – Average and Std. Dev. Stop times

<table>
<thead>
<tr>
<th>Animals</th>
<th>Average</th>
<th>Std. Dev.</th>
<th>Drugstore</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FEC</td>
<td>08.69</td>
<td>07.98</td>
<td>BM</td>
<td>14.67</td>
<td>22.55</td>
</tr>
<tr>
<td>FP</td>
<td>09.82</td>
<td>08.69</td>
<td>GEF</td>
<td>12.50</td>
<td>01.50</td>
</tr>
<tr>
<td>JA</td>
<td>09.80</td>
<td>08.67</td>
<td>HR</td>
<td>18.02</td>
<td>19.17</td>
</tr>
<tr>
<td>JPT</td>
<td>09.39</td>
<td>07.83</td>
<td>IPXL</td>
<td>16.59</td>
<td>13.92</td>
</tr>
<tr>
<td>NER</td>
<td>09.14</td>
<td>07.48</td>
<td>IPXLM</td>
<td>15.98</td>
<td>14.22</td>
</tr>
<tr>
<td>RC</td>
<td>09.43</td>
<td>15.60</td>
<td>KR</td>
<td>14.01</td>
<td>10.33</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Expedition</th>
<th>Average</th>
<th>Std. Dev.</th>
<th>Shoes</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRTC</td>
<td>09.40</td>
<td>08.78</td>
<td>FL</td>
<td>15.71</td>
<td>14.35</td>
</tr>
<tr>
<td>ESSERS</td>
<td>06.20</td>
<td>05.56</td>
<td>FLA</td>
<td>20.00</td>
<td>12.94</td>
</tr>
<tr>
<td>JLN</td>
<td>09.38</td>
<td>06.79</td>
<td>HRG</td>
<td>16.00</td>
<td>30.80</td>
</tr>
<tr>
<td>KUI</td>
<td>10.47</td>
<td>11.96</td>
<td>NEL</td>
<td>14.67</td>
<td>15.90</td>
</tr>
<tr>
<td>KV</td>
<td>14.13</td>
<td>08.64</td>
<td>SA</td>
<td>35.37</td>
<td>17.65</td>
</tr>
<tr>
<td>TPC</td>
<td>13.34</td>
<td>14.16</td>
<td>SS</td>
<td>07.99</td>
<td>35.15</td>
</tr>
<tr>
<td>ZAN</td>
<td>09.81</td>
<td>09.88</td>
<td>TAK</td>
<td>15.48</td>
<td>15.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>VH</td>
<td>18.13</td>
<td>14.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>YARI</td>
<td>19.09</td>
<td>19.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hospitality</th>
<th>Average</th>
<th>Std. Dev.</th>
<th>Retail</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>HS</td>
<td>10.92</td>
<td>09.05</td>
<td>AL</td>
<td>10.00</td>
<td>05.78</td>
</tr>
<tr>
<td>KIV</td>
<td>15.73</td>
<td>10.21</td>
<td>AMA</td>
<td>14.63</td>
<td>12.82</td>
</tr>
<tr>
<td>KKS</td>
<td>10.53</td>
<td>17.18</td>
<td>BEV</td>
<td>15.09</td>
<td>20.39</td>
</tr>
<tr>
<td>LFE</td>
<td>14.05</td>
<td>14.08</td>
<td>CB</td>
<td>13.76</td>
<td>15.65</td>
</tr>
<tr>
<td>NABT</td>
<td>18.59</td>
<td>20.61</td>
<td>FC</td>
<td>14.96</td>
<td>09.45</td>
</tr>
<tr>
<td>RAD</td>
<td>23.34</td>
<td>17.23</td>
<td>HM</td>
<td>13.44</td>
<td>11.59</td>
</tr>
<tr>
<td>SCA</td>
<td>12.91</td>
<td>13.23</td>
<td>HOG</td>
<td>11.53</td>
<td>07.44</td>
</tr>
<tr>
<td>SWG</td>
<td>14.14</td>
<td>09.90</td>
<td>KI</td>
<td>16.17</td>
<td>11.83</td>
</tr>
<tr>
<td>VERB</td>
<td>12.36</td>
<td>08.74</td>
<td>LAR</td>
<td>10.70</td>
<td>10.52</td>
</tr>
<tr>
<td>ZUID</td>
<td>12.70</td>
<td>10.35</td>
<td>MLO</td>
<td>09.67</td>
<td>09.72</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NABR</td>
<td>12.94</td>
<td>14.70</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>OPZEE</td>
<td>17.54</td>
<td>20.26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other</th>
<th>Average</th>
<th>Std. Dev.</th>
<th>Other</th>
<th>Average</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALA</td>
<td>00.00</td>
<td>00.00</td>
<td>PAT</td>
<td>12.25</td>
<td>12.02</td>
</tr>
<tr>
<td>ALR</td>
<td>15.22</td>
<td>11.06</td>
<td>POR</td>
<td>15.46</td>
<td>20.49</td>
</tr>
<tr>
<td>BAXL</td>
<td>08.67</td>
<td>10.30</td>
<td>RIJ</td>
<td>16.00</td>
<td>00.00</td>
</tr>
<tr>
<td>BE</td>
<td>20.39</td>
<td>31.41</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BER</td>
<td>10.20</td>
<td>08.73</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FF</td>
<td>26.00</td>
<td>00.00</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>