Prediction **Instrument** Development for *complex* domains

— SJOERD VAN DER SPOEL —

PREDICTION INSTRUMENT DEVELOPMENT FOR COMPLEX DOMAINS

Sjoerd van der Spoel

Graduation committee

Prof.dr. Th.A.J. Toonen, chairman & secretary	University of Twente, NL
Prof.dr. J. van Hillegersberg, promotor	University of Twente, NL
Dr. C. Amrit, co-promotor	University of Twente, NL
Prof.dr. W.H.M. Zijm	University of Twente, NL
Prof.dr. M.E. Iacob	University of Twente, NL
Prof.dr. B.M.M. Baesens	KU Leuven, BE
Dr. J. Benitez	University of Granada, ES
Prof.dr. R.J. Wieringa	University of Twente, NL
Prof.dr. W.J.A.M. van den Heuvel	University of Tilburg, NL

This research was supported by

CTIT

CTIT Ph.D. Thesis Series No. 16–400
Centre for Telematics and Information Technology
P.O. Box 217, 7500AE
Enschede, The Netherlands.

This research was conducted in part within the 4C4More project supported by Dinalog, the Dutch Institute for Advanced Logistics.

ISBN: 978–90–365–4174–9
ISSN: 1381-3617 (CTIT Ph.D. Thesis Series No. 16–400)
DOI: 10.3990/1.9789036541749
URL: <http://dx.doi.org/10.3990/1.9789036541749>
Printer: Ipskamp Printing

Copyright © 2016 Sjoerd van der Spoel

PREDICTION INSTRUMENT DEVELOPMENT FOR COMPLEX DOMAINS

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof.dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op woensdag 14 september om 16.45 uur

door

Sjoerd Jurrian van der Spoel
geboren op 7 november 1986
te Almelo, Nederland

This dissertation has been approved by:

Prof.dr. J. van Hillegersberg, promotor

Dr. C. Amrit, co-promotor

Copyright © 2016 Sjoerd van der Spoel

ISBN: 978-90-365-4174-9

Contents

Part I – Introduction

1	Introduction	14
1.1	Research questions	15
1.2	Theoretical background	18
1.2.1	Soft & hard factors	21
1.2.2	Domain analysis techniques	23
1.2.3	A review of domain analysis in complex domains	24
1.2.4	Domain intelligence	27
1.3	Terminology	31
1.3.1	Predictive models & prediction instruments	31
1.3.2	Domain, domain complexity and domain analysis	33
1.4	Structure	34
2	Methodology	38
2.1	Requirements	39
2.2	Prediction instrument development for complex domains	41
2.2.1	Preparation stage	42
2.2.2	Stage I: Qualitative assumptions	45
2.2.3	Stage II: Predictive modeling	50
2.2.4	Stage III: Model convergence	54
2.3	Prediction instrument development with soft-exclusive domain analysis	54
2.3.1	Stage I: Assumptions	55
2.3.2	Stage II: Predictive modeling	56
2.3.3	Stage III: Model convergence	57

Part II – Case studies

3	Soft-exclusive domain analysis for a simple domain	58
3.1	Case description, assumptions, and simulation construction . . .	59
3.1.1	Simulated terminal	61
3.2	Stage I: Assumptions	67
3.2.1	Goal definition	67
3.2.2	Data constraint	68
3.3	Stage II: Predictive modeling	69
3.3.1	Data selection strategies	69
3.3.2	Exploratory data analysis	76
3.3.3	Technique selection	76
3.3.4	Model training	77
3.4	Discussion	78
3.5	Conclusion	82
4	Soft-exclusive domain analysis for a complex domain	84
4.1	Case description	85
4.2	Stage I: Assumptions	88
4.2.1	Goal definition	88
4.2.2	Literature Review	88
4.2.3	Survey results	93
4.2.4	Data constraint	97
4.3	Stage II: Predictive modeling	98
4.3.1	Data selection & cleaning strategies	98
4.3.2	Reduction by data constraint	101
4.3.3	Exploratory data analysis	102
4.3.4	Technique selection	103
4.3.5	Evaluation, validation & model selection	104
4.4	Stage III: Model convergence	108
4.5	Discussion	108
4.6	Conclusion	110
5	Soft-inclusive domain analysis for a complex domain	112
5.1	Preparation	113
5.1.1	Problem identification	113
5.1.2	Expert selection	113
5.2	Stage I	114

5.2.1	Hypothesis divergence	114
5.2.2	Hypothesis convergence	121
5.2.3	Constraint definition	122
5.3	Stage II	126
5.3.1	Data selection & cleaning strategies	126
5.3.2	Reduction by data & domain constraints	129
5.3.3	Exploratory data analysis	129
5.3.4	Technique & parameter selection.	137
5.3.5	Model training.	137
5.3.6	Reduction by interestingness, deployment & domain constraints.	138
5.4	Stage III	141
5.5	Discussion	142
5.6	Conclusion	143

Part III – Conclusion

6	Discussion	146
6.1	Contribution	152
7	Conclusion	156
7.1	Answers to the research questions	156
7.2	Limitations	161
7.3	Future work	162
7.3.1	Validation	163
7.3.2	Directions	163

Appendices

A	Summary	168
B	Samenvatting	172
C	Publications	176
D	Bibliography	178

E	Details of the domain analysis review	194
F	Details of the arrival time review	196
G	Distribution center survey	200
H	Distribution center inter variable correlations	202
I	Details of the turnaround time review	206
J	Domain complexity assessment	210
	J.1 Calculating domain complexity	211
K	List of journals	212
L	Concise method handbook	218
M	Design & layout	222
	M.1 About the cover	223

Preface

Had someone told me five years ago I would be pursuing a PhD, I would have told them they were crazy. Before starting work on my master thesis, the thought had never come up to spend four more years in academia. In fact, it was the complete opposite: I was positive I would find work as a consultant, business analyst, or the like.

Then came my final project, and everything changed. I found that even though I had spent five months on my final project, I had only scratched the surface of data mining and predictive analytics. The end of my final project came long before the end of my curiosity in the matter. Therefore, I set my sights on a PhD position, and was lucky enough to get one directly after getting my master's degree.

For me, the PhD process was one of ups and downs. In the first years, it was a struggle to determine the direction I should take. One thing I did know, was that whatever direction I would take, it would require getting data. As the project I was part of at the time was involved in the supply chain, that meant I would have to get data from the involved transport companies. That turned out to be a lot more difficult than expected, as these companies were hesitant to give access to what they considered their 'secrets'. This turned into a bit of a chicken and egg problem: convincing the companies to surrender data required a clear idea of what I could offer them, and getting a clear idea of what I could offer required some test data.

Eventually, this stalemate broke, as I found a distribution center company willing to collaborate. In the months before, I had been telling myself "if only I had some real-world data". After working on this real-world data, and seeing its inherent messiness, I found myself thinking "if only I had clean, simulated data" or "if only I didn't have to work with data".

However, this first case turned out to be the key to this thesis. It showed the difficulty of working with data from real-world settings, and the difficulty of deriving predictive models from those settings. Slowly but surely, a method

emerged for dealing with these real-world problems in a structured way. So, the big problem of not being able to get (good) data, but having to create predictive models with it anyway, turned out to be the right direction for my thesis.

That was one of the main learning points of PhD project: make the best out of what you have, instead of trying to go for perfection. It was very valuable to learn to deal with poor quality data and unpredictable outcomes, perhaps more valuable than working with clean and tidy data would ever have been.

All in all, the PhD process was quite a ‘ride’. There were many lessons to be learned, some of them hard, such as dealing with rejected papers. At other times, when predictions were working as planned, or papers got accepted, it seemed as if the sky were the limit.

Not only did many things happen in the office, there were also considerable events going on outside it: a move from Enschede to Amersfoort, Marije and I got married, and to save the best for last, Laura was born a little over a year ago. The latter might not always have resulted in my most alert writing, but at least it kept me motivated until the end.

Acknowledgments

This thesis is the work of one person, made with the help and support of many. I want to express my heartfelt thanks to all who made my PhD thesis possible, but I want to address a few people in particular.

First of all, my thanks go out to my supervisor, Chintan, and my supervisor, Jos. Thanks for giving me the opportunity to pursue a PhD, and for your confidence when I felt things were not going as they should. Chintan: thanks for all the laughs, thanks for your continuous stream of new ideas (that I might not all have used), and thanks for the ever more ambitious journal suggestions. I also want to thank my fellow PhD students. My office mates Abhishta, Andrej, Gréanne, Floor, Lucas, Robin, Simon, and Wouter: thank you for the conversations (some fun, some serious), frequent coffee breaks, and good office atmosphere. Thanks also to Arturo, Arviansyah, Dissa, Fabian, Guido, Nils, and Wahid, for the lunch walks and lunch discussions about everything and nothing.

My thanks go out to my parents, for their emotional support throughout the PhD project, and for their financial support throughout my pursuit of a Bachelor’s and Master’s degree. Without their support, I quite literally could not have done a PhD.

Saving the most important for last, I want to thank my wife Marije for all her support throughout this whole process. Not only did she stimulate me to try to go for a PhD, but she has done everything to keep me motivated for these four years. That was not always easy, as for every peak, there was at least a low in this period. I especially thank her for the sacrifices she had to make during the final stretch of thesis writing, when I did not have time to help her take care of our daughter.

Sjoerd van der Spoel,
Amersfoort, March 2016

1 Introduction

The development of prediction instruments depends strongly on the domain for which they make predictions. The more complex the domain, the harder it is to make good predictions, as a complex domain has many factors that influence the prediction outcome [59], which means there are more “uncontrolled factors, [and] noise”, that must be represented in the prediction instrument [127, p.562]. That means more skill and expertise are required of the developer. Often, this expertise comes in the form of implicit domain knowledge, which is difficult to gather for a developer who is new to the domain.

Domain knowledge or domain intelligence is knowledge that has a three-fold purpose:

- First, understanding the nature and origin of the data, which allows the developer to determine the quality of the data and do data cleaning;
- Second, understanding which factors in the domain influence ‘what needs predicting’, which allows the developer to determine which variables should be included in predictive models;
- Third, understanding which predictive models are usable and interesting within the domain in terms of predictive power, integration with existing infrastructure, and integration with business rules and processes.

There is a multitude of methods for prediction instrument development, such as Shmueli and Koppius’ [127] method for predictive analytics and the Cross Industry Standard Process for Data Mining (CRISP-DM) [19], that acknowledge the importance of understanding the context a prediction instrument is developed for. CRISP-DM features *Business Understanding* and *Data Understanding* steps that have a similar purpose to that described above. The same applies to Shmueli and Koppius’ method, where that purpose is achieved by the

Goal Definition, Data Collection and Study Design, Exploratory Data Analysis, and Variable Selection steps. What neither method states however, is exactly what knowledge should be gathered, or how that is to be gathered: they merely state that domain understanding should be addressed [127, p.563, p.564 19, p.18-19].

The remainder of this introduction is organized as follows. First, we present the three main research questions that are addressed throughout this thesis. Next, we present a theoretical background of the concept and importance of domain analysis. What follows is a literature review of the ‘state-of-the-art’ in terms of domain analysis for prediction instrument development, to clarify the gap that this thesis addresses. Next, we present theory for the type of intelligence that should be gathered in domain analysis. This chapter ends with an overview of the terminology used throughout the thesis and an overview of the structure of the thesis.

§1.1 Research questions

This section presents the research questions, which are the theme that runs throughout this thesis. That central theme is the role of the domain and domain analysis in prediction instrument development. Below, we describe the meaning and relevance of each research question, and where it is addressed in the thesis.

- How does domain analysis affect predictive power of a prediction instrument? **This is the first of the three main research questions of this thesis, directly linking domain analysis with the final goal of a prediction instrument: *predictive power*.** Predictive power reflects how well a prediction instrument can make predictions, and the goal of domain analysis is to reinforce the prediction instrument ‘from the ground up’, so that predictive power is maximized.

To answer this question, we must understand how domain analysis should be performed, what methods for domain analysis fit with which domains, and what aspects of domain analysis positively affect predictive performance. This question is addressed in the theoretical background presented in this chapter, and in the case studies presented in Chapters 3, 4 and 5.

- What is the relation between domain type, domain analysis type, and predictive power? **How well a prediction instrument can make predictions depends, amongst other things, on how difficult the prediction problem**

is that it is trying to solve. The prediction problem is synonymous with the domain the prediction instrument operates in (see the terminology in Section 1.3). This question is about what makes a prediction problem a difficult/complex problem, or in other words, what makes a domain a difficult domain to make predictions about. To address that question, it is necessary to categorize domains into ‘simple-to-predict’ and ‘difficult-to-predict’ domains, and find what aspects of a domain make it more or less predictable.

Besides the difficulty of the problem, another aspect that determines predictive power is how that problem is addressed. Some treatments or solutions will be more effective than others. For example, perhaps one type of analysis would give better results in certain cases than another type would, as is claimed for other fields [14]. This question is specifically about how the type of domain analysis should vary based on the type and complexity of the problem/domain. A categorization of domain type, domain complexity, and domain analysis type is described in Section 1.2, as well as a theoretical assessment of the effect these have on predictive power. The case studies present a validation of this effect.

- How should domain analysis be performed for predictive model development? This question is about taking the theoretical findings on the relation between domain type and domain analysis type and translating it to a method. This method describes how prediction instruments should be developed, based on an analysis that fits with the domain/the prediction problem the prediction instrument is intended for. As there are different types of domains in terms of complexity [59], this question is also about determining what method fits best with what amount of complexity. Chapter 2 presents requirements for such a method, and presents our method for prediction instrument development, which should maximize the predictive power of the developed prediction instrument by using a domain analysis that fits with the problem domain. The method is used and validated in Chapters 3, 4 and 5.

The main questions above are further examined by the following sub questions:

- Why should domain analysis be executed for predictive modeling? This question addresses the importance of understanding the domain for prediction instrument development, and what the effect would be of not adequately performing domain analysis. Addressed in Section 1.2.

- What types of domain analysis exist? Different researchers address domain analysis differently. This question is about categorizing different types of analysis both from a theoretical perspective as from a practical perspective, i.e. how domain analysis is executed in practice and how that fits with theory. Addressed in Section 1.2 and Section 1.2.3.
- How does each type affect predictive power? This question addresses the theoretical influence that a type of domain analysis would have on predictive power, i.e. how the intelligence or knowledge gathered in that domain analysis would benefit the prediction instrument under development. Addressed in Section 1.2.
- What domain knowledge should be captured by domain analysis? As a complement to the questions of *why* and *how* domain analysis should be used for prediction instrument development, this question asks what the findings of the domain analysis should be, and how these can be categorized. Addressed in Section 1.2.4.
 - * How do domains differ? This question addresses the fundamental components of a domain, such as complexity, that affect predictive power from a theoretical perspective. An analysis of these components is used to distinguish domains. This is addressed in Section 1.2.
 - * What are the different types of domain? The components of a domain from the previous question are grouped into *domain types*. This categorization is presented in Section 1.2 and in Appendix J. The categorization of domains into types allows to investigate the relation between the domain type, the domain analysis type, and predictive power.
- What domain analysis type fits best for each domain type? This question combines the categorization of domains into types with the categorization of domain analysis types, and finds what aspects of a domain call for what domain analysis. The goal is that the domain analysis is tailored to the domain, so that the predictive power of the prediction instrument under development is maximized. This question is addressed in Section 1.2, Chapter 2 and validated in the case study chapters.
- What is the relation between domain complexity and domain analysis type? This question fits with the previous, and specifically addresses how the difficulty of the prediction problem influences the

Figure 1.1

The domain used for a prediction instrument is represented in data through a filter. This filter can be affected by factors from the domain, such as behaviors, goals, and politics.

choice of domain analysis type. To answer the question, we investigate if and how the requirements for a domain analysis method for a simple problem differ from the requirements for a domain analysis method for difficult/complex problems.

§1.2 Theoretical background

This section presents a theoretical background for making prediction instruments for complex systems and the role domain analysis plays in that process. First, we explain the need for domain analysis for the prediction instrument development for complex domains, and what types of analysis exist. Next, we present a structured literature review, to illustrate the extent to which domain analysis is used for complex domains in practice, which serves to clarify the gap that this thesis addresses. The theoretical background ends with a presentation of the types of intelligence that are produced by this domain analysis.

Prediction instruments are developed within a particular context, such as an organization or group of organizations. This context, which is “a concept of a whole which has properties as a single-entity, so-called ‘emergent properties’”, is referred to as a *system* [22, p.278] or as a *domain*. We consider a system to be the immediate context of the prediction instrument, which is known as a *domain* in the field of data mining and predictive analytics [14, 127]. When we refer to domain in this research, that includes everything that directly *influences* or *interacts* with the prediction instrument under development.

To make predictions, a prediction instrument requires data. This is either used to train a predictive model, or it is a new observation for which the outcome is to be predicted. This data is provided by/recorded in the domain. This

Figure 1.2

Abstract domains and domain instances. A lightning symbol (⚡) signifies that an attribute is affected by noise. Here, the two domain instances also both have extra attributes, that are not shared between instances or with the abstract domain.

means that (relevant aspects) of the domain are *represented* by the data. Figure 1.1 shows this relation between the domain and the data. The representation of the domain characteristics in the data is not straightforward: it passes through a *filter*. This filter determines *what* and *how* aspects of the domain are represented in the data.

Key to domain analysis for prediction instrument development is understanding the domain and understanding how it is filtered through into data. This determines amongst others what variables to select and how to perform data cleaning, as understanding the domain leads to knowing what aspects are important, and understanding the filter leads to knowing how these are represented. Domain understanding and the resulting variable selection and data cleaning are both important steps in predictive model development methodologies such as that by Shmueli and Koppius [127] or the Cross-Industry Standard Practice for Data Mining (CRISP-DM) [19]. However, neither method clarifies how this domain understanding should be done.

Existing literature describes roughly two approaches to domain and filter understanding. The first, which is called *systems engineering*, considers each domain as an instance of an abstract and well-known domain [22, 41]. The understanding of that abstract domain can then be immediately transferred to the instantiation.

The second group of approaches, which contains *soft systems methodology* [22] and *system dynamics* [41] argues that this way of thinking is not applicable to domains where aspects such as human behavior, people trying to work together, interactions with the environment, and conflicting goals within the sub parts of the domain play a role [22, 41]. The argument is that these are too different from an abstracted variant.

Soft systems methodology emerged in response to (the deficiencies of) systems engineering at the end of the 1970s. In systems engineering soft factors such as behavior, politics, or trust are considered not relevant, and domain analysis consists of trying to model the domain mathematically. Often, that means finding a popular problem that is similar and reusing the analysis made there [22].

In soft systems methodology a domain is considered ‘a complex problematical human situation’ [22], and it states that inquiry should be made to understand each aspect of that domain by talking to those involved. It is not a replacement of systems engineering, but rather an extension, as Checkland writes: “[Soft systems methodology] derive[s] both from interpreting the facts and logic of a situation and from engaging with the myths and meanings through which other managers make sense of it” [22, p.278].

Forrester’s system dynamics agrees with Checkland’s views of rejecting the use of mathematical simplification for analyzing complex domains [41]. Forrester believes that this type of soft-exclusive or ‘static’ approach cannot capture the complexities of domains in the real world. However, Forrester states that soft system methodology’s qualitative methods of analysis lack rigor, as they depend on the researcher’s intuitions [41]. System dynamics aims to resolve this by its ‘disciplining framework’ [41, p.248] of model formulation/domain description and model testing/domain simulation. Both Forrester and Checkland state that approaches that reuse existing domain analyses are *not applicable* for complex domains [22, 41], but suggest that this is not problematic for simple domains.

For example, consider predicting truck travel time. The domain for this case is an instance of the ‘travel time’ domain, which is thoroughly covered in literature. A soft-exclusive approach would be to take this literature, and use it as a basis for variable selection and data cleaning. The most important variables would then be weather and traffic [132]. If there are in fact no other

factors that could affect the outcome, then this soft-exclusive analysis could give good results; but as previous work shows [132], if there are behavioral influences (such as that the drivers have no intention to be punctual), predictive power is severely reduced. This shows that a domain analysis from an abstract domain cannot be translated one-to-one to an instance of that domain that is subject to behavior and other ‘soft’ factors.

Figure 1.2 further illustrates this point. In the figure, there are two instances of an abstract domain. The abstract domain has a minimal set of attributes that every one of its instances also has. However, as instances represent complex domains, they are affected by other factors that only apply to that specific domain instance, such as culture, politics, and behavioral influences. Furthermore, in instances of the domain, the same attributes might be differently affected by ‘noise’, i.e. factors that disturb the domain-data filter and therefore the quality of data [134].

The domains where the aspects above play a role are called *complex domains*. Jackson and Keys provide the following definition: A complex domain is a domain where people are trying to act together; that involves politics, myths and meanings; that is unstructured; that is not fully observable; where parts have their own goals; that is probabilistic in nature; open to its environment; and subject to behavioral influences. Consequently, a domain with the opposite attributes is a simple domain [59].

In a complex domain, the filter between domain and data could be affected, adding noise or bias to the data. Data might be manipulated to be more favorable to certain actors in the domain, or mistakes could have been made in recording data. Previous work has shown several examples, such as truckers who were unwittingly incentivized to be late at a distribution center, leading to unpredictable arrival times; and a hospital where the method for data recording added considerable noise [132, 134].

§1.2.1 Soft & hard factors

Factors exist on a spectrum from hard to soft. The less influenced a factor is by the people, politics, and behavior in the domain, the more hard it is. These factors themselves influence what is to be predicted, or influence the quality of data or information that is used to make predictions. Below, we will give a definition of a factor, and what makes it ‘hard’ or ‘soft’.

We use the following definition of a factor: a phenomenon or construct from the domain that affects what is to be predicted. A factor could be a variable, or a type of noise affecting the relation between variables or between a variable and what is to be predicted, such as the types discussed in [134]. The

latter is synonymous with information quality, such as discussed in [29,33]. If such a factor is directly or indirectly influenced by behavior, politics and culture, it is considered a soft factor.

Politics can be defined as *power in action*. Power is the ability to get someone to do they would otherwise not do, or the ability to prevent them from doing something they want to do [94]. Politics are taken to become relevant in circumstances where decisions are “subject to negotiation, contestable, [or] mouldable” [94, p.85], and/or when a domain involves individuals or groups with different motivations; when there are different views on “the state of the world, and what the future holds” [94, p.85]; or when conditions in the domain are uncertain.

Davenport *et al.* [33] describe *information politics* as an example of a soft factor that affects information quality. These information politics are about how parts of the organization, that all have their own “culture, leaders, and even vocabulary” [33, p.53] share or withhold information. Trust is another soft factor that affects information quality [29]. Chepaitis found a Soviet-era prejudice against data sharing, “rooted in the belief that power and security are eroded if data is shared” [29, p.55], that was strengthened by a fear that revealing assets (through sharing data) to the government could result in higher taxes.

As mentioned above, the distinction hard–soft is not binary, but rather a spectrum. Weather could be a hard factor influencing arrival time: it is not subject to behavior or politics. On the other end of the spectrum, ‘motivation’ is a soft factor: it directly reflects a behavioral aspect. In between are combinations, i.e. factors that are neither free from behavioral influence nor a direct reflection of behavior. Congestion could be an example, as it is not necessarily a direct consequence of behavior, but behavior could change its effect on arrival time: some drivers are more adept at avoiding traffic jams.

A soft factor is a factor that is unique to one domain, they represent “unique idiosyncracies which characterise human situations” [22, p.275]. Although the same factor could be at play in two domains, its effect will be (slightly) different in each domain. For example, trust is a factor that plays a role in many domains, but the strength of its effect differs. For example, research on the relation of trust and information quality shows that the lack of trust between organizations in Russia leads to a lack of information quality [29]. The author of this study found that Russian organizations in the post-Soviet years were hesitant to share information, as they were afraid this would lead to higher taxes [29]. This is specific to this context (post-Soviet Russia), and probably not even the same for each organization: the author found that there was more trust in some regions than in others.

Similarly, Davenport *et al.* state that the influence of information politics depends per domain. Davenport *et al.* describe five different models of information politics, that all have a different impact on information quality. For example, the ‘technocratical utopianism’ model leads to low information quality, whereas ‘federalism’ is thought to lead to higher information quality [33]. Therefore, it is necessary to study the domain in detail, to assess which model is in use, and in extension, what should be expected in terms of information quality.

The same could be the case for more ‘hard’ factors, such as the congestion discussed above. In some domains, drivers could be motivated to arrive as early as possible, urging them to avoid traffic jams, whilst in others (see Chapter 4), drivers could lack that incentive.

§1.2.2 Domain analysis techniques

We refer to an analysis that includes soft factors as soft-inclusive, and by extension, an analysis that excludes soft factors, i.e. only takes those into account at the ‘hard’ end of the spectrum, is soft-exclusive. We also consider an analysis that does not take the *domain-specific* effects of a soft factor into account as soft-exclusive: an analysis that does not consider how the behavior, culture and politics specific to a domain affect what is to be predicted can be considered the same as an analysis that does not consider these factors at all.

For a soft-exclusive domain analysis, a good candidate technique is a literature review. When attention is paid to completeness, this will reveal the relevant aspects of the group of domains the domain belongs to. If the domain is simple, then these aspects would be translatable from the abstract domain to the specific domain, and it is less likely that relevant aspects of the simple domain are missed.

Even though a soft-inclusive domain analysis approach is widely considered important for predictive modeling [14, 51, 126, 127] there are few methods that make domain analysis a clear, rigorous, and transparent process. Often, domain understanding is at best an implicit part of model development. This could either mean that it was not necessary, because the developer is an expert on the domain, or that it was not done. Regardless, it seems there is a demand for transparent, repeatable techniques that capture relevant knowledge from the domain for complex domains.

§1.2.3 A review of domain analysis in complex domains

In the previous section, we have described the purpose of domain analysis in prediction instrument development. To find how domain analysis, and especially to find what type of analysis (soft-inclusive or soft-exclusive) is used for simple and complex domains, we have performed a literature review. We base our review method on that of Webster and Watson [155].

For our literature review, we focus on papers about predictive modeling for the supply chain. We have chosen this focus, as the supply chain is a known complex field. Therefore, a review of papers from this field should provide a good overview of the domain analysis methods that are used in practice.

Search & selection strategy First, we established a list of the fields related to predictive analytics and the supply chain: information systems, artificial intelligence, interdisciplinary applications of computer science, industrial engineering, operations research, supply chain, decision sciences, business and management.

For each field, a top twenty list of journals was selected based on journal impact and other measures of academic relevance. This selection is based on the AIS Senior Scholars' Basket of Journals [5], the Journal Citation Report of the ISI [144], the University of Sydney Journal Rankings [54], SCImago [122] and several peer reviewed scientometric publications on journal rankings [45, 92, 93, 100, 123, 138]. We used only journals, as we are interested in those papers that are considered the best in this field, and journals are the field's primary type of contribution.

Using the list of journals, we executed the following query on Scopus and Web of Science: (*"supply chain"* OR *scm* OR *logistics*) AND (*analy** OR *"big data"* OR *"data mining"* OR *"business intelligence"* OR *"decision support"*) AND *predict**, resulting in 626 papers.

The abstract inclusion criteria are the following: they are related to one or more aspects of the supply chain, and they mention some type of prediction using data. Two reviewers analyzed the abstracts using these criteria and decided for each whether it was highly relevant. Comparing these decisions, the reviewers had sufficient inter-rater-reliability (Cronbach's α :0.80, Krippendorff's α :0.67).

In the final round of review, we reviewed the full text of the highly relevant papers. The inclusion criteria are that the paper discusses prediction related to the supply chain, and that empirical data was used to make those predictions. After this round, 47 papers remained.

Figure 1.3

Results of the literature review and domain complexity analysis. Table E.1 on page 194 shows the papers we found for each domain and analysis type.

Next, we investigated the complexity of the domains in each paper, using an assessment tool based on Jackson and Keys' definition of complexity [59]. The assessment tool, which consists of eight questions, is in Appendix J, including a formula to calculate a complexity score. One author answered the questions for all 47 papers. Another researcher did the same for a randomly selected subset of 6 papers to rule out rater bias. Sufficient inter-rater reliability (Cronbach's α : 0.87, Krippendorff's α : 0.72) indicates that the reviewers were in agreement.

Results Figure 1.3 shows the results of the literature review. As mentioned in the previous section, we ranked each paper's domain complexity. Papers that had a complexity score > 3.00 are categorized to deal with a *complex* domain, papers with a score ≤ 3.00 are about *simple* domains, as this is halfway between *simple* and *complex*. Figure 1.4 shows how domain complexity is distributed for all papers in the review. Similarly, we looked at each paper's domain analysis approach. Papers that use soft analysis for variable selection or data cleaning are labeled soft-inclusive, the others are labeled soft-exclusive.

Figure 1.4

Complexity of the papers in the literature review. All papers with a complexity of 3 or above are considered complex papers, where 1 is the lowest possible complexity score and five is the highest possible complexity score. As the figure shows, there are no papers with a complexity between 1.5 and 3.5. See Appendix J for the calculation of the complexity score.

The majority of papers (96 % of the total number in the review) deals with a complex domain. These papers deal with real-world case studies in the supply chain, ranging from scheduling problems to predicting macro-economy figures. The two papers that deal with a simple domain both make predictions about a domain made up of only intelligent agents (which are not subject to behavioral influences).

87% (41/47) of the total papers and 91% (41/45) of the papers dealing with complex domains use a soft-exclusive domain analysis. Mostly, these papers use literature review as a means of variable selection, if any is explicitly mentioned at all, or use a data analysis as a substitute for domain analysis. This means that variables are selected based on the provided data. Data cleaning is mentioned in only seven of the papers.

Most papers in complex domains that use soft-exclusive domain analysis are examples of Operations Research. The methodology of these papers tends to be to view cases as instances of well-known problems, resulting in a domain analysis based on literature about the well-known problem. In these papers, the artefact developed is usually a mathematical model, that is primarily used to increase efficiency (e.g. optimizing the use of resources), but is also used to make predictions. These predictions usually are the amount of efficiency increase if the developed mathematical model is used, but could also be forecasts.

The six papers that did include soft factors mostly used interviews as a means of capturing domain intelligence. Rabelo *et al.* used Forrester's system dynamics, and created a model of a supply chain based both on existing theory and interviews with those involved [116].

§1.2.4 Domain intelligence

The paragraphs above discuss *why* a soft-inclusive domain analysis fits with more complex domains. That leaves open *what* exactly should be analyzed and *how* that analysis should be made in the specific context of prediction instrument development.

Cao *et al.* [14] provide an answer to both the *why* and *what* question, in their Domain Driven Data Mining (D3M) method. Although their method is focused on *understanding* the domain rather than making *predictions* about the domain, it can be adapted to be suited for prediction purposes. The main argument presented for D3M is that *data does not exist in a vacuum*, and that understanding the domain of the data and gathering intelligence is key to getting valuable insights from data [14].

Figure 1.5

The six types of domain intelligence that need to be captured during domain analysis [\[14\]](#). Organizational, social, and human intelligence have considerable overlap, just as data and network intelligence. In turn, domain intelligence encompasses and extends all five intelligences.

The *what* is that there are six types of intelligence that should be gathered in the domain analysis part of prediction instrument development: data intelligence, network intelligence, social intelligence, human intelligence, organizational intelligence, and domain intelligence [14]. The relation between these types of intelligence is shown in Figure 1.5.

Data intelligence shows the volume, variety, velocity and veracity of the available data in the domain. It is closely related to *network intelligence*, which shows how that data is distributed physically and shows how data sources are linked.

Human intelligence is comprised of the prediction outcome related thoughts, experiences and beliefs of people involved in the domain. *Social intelligence* reveals what groups exist within the domain, what the dynamics within those groups are, and how they are related. *Organizational intelligence* reveals roles and structures within organizations in the domain.

Domain intelligence is the superset of these five types of intelligence, and covers any type of intelligence from the domain. Besides the five types mentioned above, this includes other factors such as laws, business rules and business practices. All types of intelligence are synthesized into constraints, that determine which predictive models will be *actionable*: i.e. interesting and usable [14].

Constraints As mentioned in the description of *domain intelligence*, one product of gathering these intelligences are constraints [14, p.20] on the prediction instrument that is to be developed. To understand these constraints, it is important to understand what Cao *et al.* call *actionability*, or how usable and useful a prediction instrument will be as a practical tool for decision support. Below, we discuss the constraints and how they are applied in prediction instrument development.

There are four dimensions constraining the actionable prediction instrument space. The *data constraint* relates to the quantity, quality and structure of the available data. Any instruments that need data that is not available in terms of any of these three attributes, need not be developed. For example: the intelligence gathering might reveal that ‘mood’ is an important predictor, but if this is not available in the data, any prediction instrument that relies on it would not be usable.

The *interestingness constraint* relates to the performance of the prediction instrument. If the performance, such as accuracy, precision is too low, or error of the predictions is too low (for accuracy and precision) or too high (for the error), the instrument cannot be used for decision support.

Figure 1.6

Prediction instrument space and *actionable* prediction instrument space. The left side of the image shows one constraint on the y -axis with upper bound c'_y and lower bound c_y , and similar for the x axis with lower bound c_x and c'_x . The area within these bounds is the actionable instrument space, colored in white. As the right hand side of the image shows, this two-dimensional space is further bound by constraints on the z -axis. (For completeness sake: there are in fact four constraint dimension, but the w -axis is not pictured for reasons of clarity.) Constraint dimensions can be either quantitative or qualitative.

The *deployment constraint* relates to the integration of the prediction instrument with existing infrastructure. Instruments that are not supported by existing hardware and software; cannot be integrated with existing decision support systems; or cannot give predictions in real time are (depending on context) not usable for decision support.

Finally, the *domain constraint* relates to other constraints from the domain where the instrument will be used. These include regulatory measures, integration with existing business processes and compliance with business rules. The domain constraint also determines which performance metric is to be used, and what type of predictions (e.g. classification or regression) the prediction instrument should make.

Consider the space of all possible prediction instruments that could be developed based on the intelligence gathered *predictive instrument space*. For each type of constraint (four in total) there is one dimension in this space, which is limited by the constraint. Combining all four constraint dimensions results in the *actionable* area of prediction instrument space. If a prediction instrument falls outside the limits of one or more constraints, it can no longer be considered actionable. Figure 1.6 shows the prediction instrument space in three dimensions, and how the constraints determine which area of that space is actionable.

§1.3 Terminology

In this section, we explain the terminology used throughout the thesis. The purpose of the section is to both clarify new terminology and to explain how we use existing terminology. The terms covered in this section are predictive models, prediction instruments, domain types, and domain analysis types, of which prediction instrument is the main new term introduced.

§1.3.1 Predictive models & prediction instruments

A *predictive model* takes a set of observations and outputs a prediction, i.e. an outcome for the new observation. This prediction can be an interval, a distribution, or an exact value, and any method that can make predictions is a predictive model [126, p.292]. Any method that can *create* predictive models is what we call a predictive modeling *technique* or *algorithm*, as it follows a set of procedures to create a predictive model from data.

There are many techniques, that all have their own strengths and weaknesses. A well-known example of a technique is the Classification and Regression Tree method [12], part of the decision tree group of techniques. This

Column representing *the goal variable*

↓

Row	Age	Gender	Savings	Mortgage
1	19	M	€ 5000	False
2	39	F	€ 15000	True
3	22	M	€ 15000	True
⋮	⋮	⋮	⋮	⋮
n	45	F	€ 10000	False

↑ ↑ ↑ ↑ ↑

Columns representing *variables*

Row representing *instances*

Figure 1.7

An example dataset, where the variables are represented by columns, and the instances (or observations) are represented by rows. In this example *Mortgage* is the goal variable. So, when values for each of the other variables are put into the predictive model, it outputs a value for the goal variable.

method takes a historical dataset consisting of both the independent variables and the variable that is to be predicted, i.e. the dependent variable. It then determines which values for the independent variables best (most cleanly and completely) split the dependent variable into parts (such as intervals or exact values).

Another group of techniques is called ensemble techniques, of which AdaBoost [43, 44] and Random Forests [11] are examples. These techniques take a group (hence ensemble) of other (weaker) predictive models, and considers the prediction of each of those a *vote*. The ‘popular’ vote determines the prediction of the ensemble.

A key differentiation is that between a *technique* and a *model*. A technique is used to create a predictive model, like a decision tree or ensemble. A predictive model is used to make *predictions* for new observations.

A prediction instrument is a combination of an predictive model with the technique and technique parameters that were used to create it, a data selection and refinement strategy and a set of constraints necessary to make the predictions the business needs.

Data used in a prediction instrument can be represented by rows and columns of values. Rows represent instances, or observations, and columns represent variables. Figure 1.7 gives a graphical example of a dataset and the instances and variables it contains. One of the columns represents the goal

variable, i.e. the variable the instrument will predict. Of course, this column will only have a value for historical/existing rows, as it must be predicted for new observations (i.e., rows in the matrix representation of a dataset). The *data selection and refinement strategy* is a definition of what data should be used (which rows and columns in the matrix) and how this data is to be processed and cleaned.

A number of rows (determined by an algorithm-specific parameter) of the data called the *training set*, which includes the goal variable column, is fed into the algorithm to create a *model*, which could be a neural network, a decision tree, a regression model, etc (see *Predictive model*).

New rows, i.e. rows that do not have a value for the goal variable column are fed into the model, which then produces the prediction: a value for the goal variable for that row¹.

The constraints determine whether the predictive model is useful in practice. There are four constraints: a *data* constraint, which states what data is available and can be used; an *interestingness* constraint, which states how well the model should predict; a *deployment* constraint, which states the existing infrastructure the model should integrate with; and a *domain* constraint, which states other limits on usable models, such as laws and business rules. These constraints are presented in Section 1.2.4.

§1.3.2 Domain, domain complexity and domain analysis

Prediction instruments are created to solve a prediction problem. The context of this problem, which is also the context of the prediction instrument under development, is “a concept of a whole that has ‘emergent properties’” [22, p.278]. In the original source of this definition, this is what is called a *system*, but we refer to it as a *domain*, as this is the more generally used term in the field [14, 127].

The domain contains all people, organizations, and other interconnected elements together forming a whole that influences what is to be predicted. One of the emergent properties of the domain as a whole is the goal variable of the prediction: all actions and resulting data in the domain could affect the prediction goal.

Domain complexity is a measure of the *amount* and *type* of factors in the domain that affect the goal variable or affect the quality of data. We use Jackson and Keys’ definition of complexity [59], which is built up as a checklist with

¹This way the model can also be tested, by feeding in rows for which the goal variable value is known. The predictions can then be compared to the actual values to determine the performance of the model.

five items: a domain is complex if it is not fully observable; if it is probabilistic in nature; if it is open to its environment; if its parts have goals of their own; and if it is subject to behavioral influences and political, cultural and ethical factors. Likewise, there is a checklist for a simple domain: it is fully observable; it is deterministic in nature; it is closed to its environment; its parts are passive and have no goals of their own; and it is not subject to behavioral influences.

Domain analysis is the process of gathering knowledge or intelligence for the purpose of better understanding the domain and how it affects the goal variable. This includes understanding which variables to select and assessing the quality of data [127]. The latter allows to clean up data, and in turn to assess whether there is enough data of sufficient quality to create a prediction instrument.

A simple form of domain analysis is a review of similar domains. Similar means that they are domains that deal with the same prediction problem, such as ‘predicting arrival time’ or ‘predicting customer churn’. In other words, they are two instances of one abstract domain. The review would result in variable selection and data cleaning strategies that have been used previously, which are then repurposed for the new domain.

This approach of domain analysis would not reflect domain-specific factors and conditions that affect the goal variable, such as culture, politics, behavior, or ethics, but would likely only include those variables that are directly quantifiable. As this type of domain analysis excludes the soft, domain-specific factors, or the soft aspects that influence these (quantifiable) variables, we refer to it as a *soft-exclusive* domain analysis.

A complementary approach is to try to capture these soft factors, in addition to gathering theoretical understanding from a literature review. In such an approach, the prediction instrument developer would investigate the domain and those involved directly, by field study, interviewing, or other means. This type of analysis, which both covers the abstract domain and captures soft, domain-specific factors is what we refer to as a *soft-inclusive* domain analysis.

§1.4 Structure

This section gives an outline of the remainder of the thesis. This chapter has discussed a theoretical background on domain type, domain analysis, and their relation to predictive power. In so doing, the research questions that involve the *why* and *what* of domain analysis as part of prediction instrument development have been addressed from a theoretical perspective.

The theoretical background in this chapter outlines a hypothesis on how domain analysis affects predictive power. The hypothesis is that a prediction instrument developed for a *complex domain* (that is affected by behavioral influences, open to its environment, partially observable, and has parts with their own goals) with *soft-inclusive* domain analysis that takes behavioral influences, politics, sub-part goals and its environment into account will have *higher predictive power* than a prediction instrument developed for the same domain with *soft-exclusive* domain analysis. The complementary hypothesis is that a domain analysis method that does not look for behavioral influences and the like (a soft-exclusive domain analysis) is a good (enough) fit for a simple domain/simple prediction problem.

Chapter 2 introduces requirements for a prediction instrument development method that uses soft-inclusive domain analysis. The requirements are then used to create our method: *Prediction Instrument Development for Complex Domains*. The chapter also describes a prediction instrument development method with soft-exclusive domain analysis, based on Shmueli and Koppius method for predictive analytics and Webster and Watson's structured literature review method [126, 127, 155](#).

Both methods are validated in case studies described in Chapters 3, 4, and 5. These case studies serve not only as validation of the method, i.e. its applicability in practice, but also as validation for the hypotheses above. Chapter 3 demonstrates the use of the soft-exclusive method for a simple domain, to show that there is indeed a good fit, i.e. sufficient predictive power. To ensure the simplicity of the domain and that it is not influenced by any 'soft' factor, it is fully artificial: the domain is a simulation of a container terminal.

Chapter 4 illustrates the effect of using a soft-exclusive domain analysis for a complex, real-world domain. In this case study, which presents the problem of predicting truck arrival time at a distribution center, we demonstrate a domain analysis based solely on existing literature, that does not take into account that parts of the domain have goals of their own, that there are behavioral influences, and that the domain is not fully observable, and we discuss how that affects the predictive power of the resulting prediction instrument.

In the final case study, presented in Chapter 5, we directly compare the soft-inclusive and soft-exclusive domain analysis methods for a complex domain. The domain discussed in this paper is that of a real-world container terminal in the Port of Rotterdam, so that its results can be compared with those from Chapter 3, and its simulated domain. In this chapter, the soft-inclusive and soft-exclusive method are both tested, to see if the hypothesized significant difference between soft-inclusive and soft-exclusive methods manifests itself in practice.

		<i>Domain type</i>	
		Simple	Complex
<i>Analysis type</i>	Soft-exclusive	Chapter 3	Chapter 4, 5
	Soft-inclusive		Chapter 5

Table 1.1

An overview of the case studies in this thesis. The case studies cover the meaningful combinations of the two types of domain (simple and complex) and the two types of domain analysis (soft-inclusive and soft-exclusive). This means that there is no case study using soft-inclusive domain analysis for a simple domain.

Table 1.1 gives an overview of all three case studies. The figure shows the domain type as well as the method used for each case study. There is no case study for a soft-inclusive domain analysis of a simple domain, as theory suggests that a simple domain is unaffected by the factors a soft-inclusive domain analysis aims to capture.

The discussion in Chapter 6 compares the results of all three case studies, and discusses the theoretical and practical contributions of this thesis. In particular, the discussion focuses on the merits of the soft-inclusive prediction instrument development method presented in Chapter 2, and the extent to which the two hypotheses presented above are validated by the case studies.

In Chapter 7, which concludes this thesis, the answers to the research questions from Section 1.1 are summarized. Besides answers to the research questions, the conclusion presents limitations to the work presented here, as well as directions for future research on domain analysis for prediction instrument development.

2 Methodology

This chapter builds on the theory presented in the previous chapter by giving methods for prediction instrument development. The first and main method presented here uses a soft-inclusive domain analysis, which takes behavior, politics, ethics, culture and (conflicting) goals of parts of the domain into account. The second method, which is used to contrast with the first, uses a domain analysis that only takes more easily quantifiable factors into account, and since it does not account for any of the soft factors mentioned, it is called a soft-exclusive domain analysis.

The chapter starts with requirements for a prediction instrument development method with soft-inclusive domain analysis, based on a review of literature related to domain analysis, design science, and IS methodologies. There are a total of five requirements for such a method, which are used to create our method: *prediction instrument development for complex domains* (PID-CD), which incorporates a soft-inclusive domain analysis.

Section 2.2 describes the PID-CD method. The method consists of three main stages: first hypotheses on the domain are gathered with the help of domain experts, then predictive models are developed from those hypotheses, and finally one predictive model is selected and made into a prediction instrument.

Section 2.3 describes what a prediction instrument development method with soft-exclusive domain analysis would look like. Like PID-CD, this method is built up of three stages: first literature is surveyed for hypotheses, which are then converted to predictive models. The best performing predictive model, combined with the technique, parameters, and domain intelligence used to create it, is made into a prediction instrument.

§2.1 Requirements for a prediction instrument development method

The theoretical background makes it clear that soft-inclusive domain analysis fits with complex domains. The results from the literature review in Section 1.2.3 make it clear what type of analysis is in fact used: the overwhelming majority of papers that deal with complex domains mention only the use of soft-exclusive methods for domain analysis. There is a considerable misalignment between the type of analysis that should theoretically be used for complex domains, and that which is used in practice.

However, all papers in the review have reasonable predictive performance, which shows that the researchers in fact did have a proper understanding of the domain they were working on. As domain analysis is given at most brief attention in these papers, we therefore assume that a soft-inclusive domain analysis was used, just not *explicitly*. For example, if the researcher is an expert in the domain, then the process of gathering that expertise might not be mentioned in the paper. In other words, the domain analysis is left opaque, and only the quality of the produced predictive model shows that the researchers fully grasp the complexities of the domain.

As the theoretical background shows the importance of soft-inclusive analysis, the gap that should be addressed is how this analysis should be performed. To address this gap, we present requirements below for a transparent prediction instrument development method with soft-inclusive domain analysis. The requirements are based on literature related to method and artefact development, as well as the literature discussed in the theoretical background presented in Chapter 1.

Design science approach The goal of the method is to design and develop an artefact: the *prediction instrument*. The field of design science research provides general methodologies on how to address such a goal [50, 108]. The prediction instrument development method should follow these good practices, and consist of the same basic steps: *problem identification* and *objective definition*, which form the domain analysis part of the method; *design and development*, which consists of creating predictive models; *demonstration* and *evaluation*, which consists of testing the predictive power of the predictive models; and *communication*, which consists of combining the best predictive model and the technique, parameters, and domain intelligence used to create it into a prediction instrument [108].

Quantitative and qualitative methods The domain analysis part of the method should capture soft factors, which fits with the use of qualitative methods such as interviewing. In contrast, the development of predictive models uses quantitative methods such as statistical analysis and machine learning. Therefore, the combination of these two parts, the prediction instrument development method, is an example of *mixed methods* research [153]. In prediction instrument development, the quantitative methods are used to validate the qualitative methods: predictive models are created to test which factors from the domain analysis truly affect the prediction outcome. It could also be said that the quantitative predictive models extend the knowledge of the domain by showing which factors affect what is predicted. This type of relation between qualitative and quantitative methods (or mixed method purpose) is called *extension* or *confirmation* [153].

Domain-driven For complex systems, the development of prediction instruments should start with domain analysis. Next, amongst other uses, the findings from the domain analysis are used for variable selection and data cleaning, which are subsequently used for predictive model development. So, the prediction instrument development is driven by the domain, rather than by data ¹.

There are six types of intelligence that should be gathered in the domain analysis part of prediction instrument development: data intelligence, network intelligence, social intelligence, human intelligence, organization intelligence, and domain intelligence [14]. The six types of intelligence are explained in Section 1.2.4. The relation between these types of intelligence is shown in Figure 1.5.

There are four constraints²: a *data constraint*, which states the quantity and quality of available data, and therefore rules out predictive models that require more or better data; an *interestingness constraint*, which states the minimum performance needed from the model; a *deployment constraint*, which determines what infrastructure predictive models have to be able to interact with; and a *domain constraint*, which contains laws and business rules the predictive model should be in accordance with, but also determines what performance measure to use, and any other reason why a predictive model would not be actionable.

¹Purely data driven predictive model development does not fit with complex systems, as it depends on the quality of the filter between domain and data (see Section 1.2 and Figure 1.1): finding out whether the data is manipulated or noisy from that same data is difficult if not impossible.

²See Section 1.2.4 for a more detailed description of constraints, actionability, and their relation

Builds on existing methodologies This is not a stringent requirement, but given that there is a multitude of methodologies for predictive model development, such as that by Shmueli and Koppius [126, 127], and more generically for data mining, the Cross Industry Standard Process for Data Mining [19], making use of these established methods ensures the rigor and validity of our method. Using parts of an existing method ensures greater validity and generalizability. These methodologies represent best practices for predictive model development, and should therefore be included in the prediction instrument development method.

Efficiency As a final requirement, the prediction instrument development method should be efficient. As it replaces implicit domain analysis or other opaque methods of gathering domain intelligence, it should provide as little burden as possible. This means that no unnecessary predictive models should be developed.

§2.2 Prediction instrument development for complex domains

In this section, we describe our method for prediction instrument development that captures and uses soft factors from the domain, called *prediction instrument development for complex domains* (PID-CD). Our method is based on intelligence meta-synthesis, which has previously been used to create economic models [51] and as an analysis tool for association rule mining [14]. The name derives from the fact that it synthesizes new intelligence (in the form of a model) from domain intelligence. As this method gathers and uses intelligence to create intelligence, the authors consider it ‘meta’. Note that this is not how this term is generally used in modeling information systems: there, meta-modeling is about describing the constructs needed to create models. However, as IMS is the term used by Gu and Tang, and their method forms the foundation of our method, we will use their term throughout this chapter.

Below, we describe how we have adapted the method for the purpose of prediction instrument development.

The method consists of three stages. In stage I, *qualitative assumptions* are gathered from the domain. For prediction instrument development, these would be assumptions or hypotheses on factors influencing what is predicted. The stage starts with a field study, brainstorm or interview with domain experts to determine all relevant hypotheses. This is called *hypothesis diver-*

gence. The unique and interesting hypotheses are then determined in the *hypothesis convergence* step. In stage I, the domain experts are also consulted to determine the five domain constraints.

The whole of stage II is based on Shmueli and Koppius' method for predictive analytics [126, 127], and consists of variable selection, data cleaning, technique selection, and evaluation. In stage II, predictive models are developed based on the domain analysis from stage I. The hypotheses are used as a basis for variable selection and data cleaning.

The data and domain constraints are used to determine which hypotheses can be tested (a hypothesis can only be tested if there is enough data of sufficient quality available). Next, using selected techniques, predictive models are created. These are then evaluated to determine their predictive performance relative to the interestingness constraint.

In stage III, the predictive models from stage II are validated qualitatively. The models that have sufficient performance and that do not violate the domain or deployment constraints are presented to the decision makers from the domain. Then, either by discussion, voting, or a technique like the Analytic Hierarchy Process [121], one predictive model is selected. If the decision makers decide that no predictive model is good enough for whatever reason, or all predictive models were eliminated by the constraints, parts of the PID-CD process can be repeated. If a predictive model is selected at this stage, that forms the basis for the prediction instrument, combined with the constraints, technique, and data selection and cleaning strategy that were used to create it.

Table 2.1 shows each of the requirements for a prediction instrument development method that were discussed above, and how these are covered by PID-CD. As the table shows, PID-CD is a complete fit with the requirements.

Before the three core stages of PID-CD, there are two initial steps: problem identification and expert search and selection. All five stages are described in detail below. An overview of the sequence of the steps in PID-CD and what their in- and outputs are is shown in Figure 2.4 on page 53.

§2.2.1 Preparation stage

Problem identification Preparation for PID-CD starts with an assessment of the problem domain and defining the prediction goal ('what needs predicting' [127, p. 562]). Commonly, prediction is about giving an outcome value for new observations [127]. Shmueli and Koppius suggest that when defining the goal, the type of outcome required should also be determined (e.g. a categorical or numerical value), as this influences how models are developed later on. In

<i>Requirement</i>	<i>Reflected by</i>
The method covers <i>problem identification, objective definition, design and development, demonstration, evaluation, and communication</i> .	The first two are covered by PID-CD stage I. Design and development is covered by stage II. Demonstration and evaluation are covered by both stages II and III. Stage III covers communication.
The method makes use of mixed methods, where quantitative methods are used to confirm and/or extend findings from qualitative methods.	Stage I develops assumptions about the domain through interviewing and field studies. These are then validated by quantitative predictive models in stage II. In stage III, decision makers from the domain decide which predictive model to use, which is a form of qualitative validation.
The method captures data, network, social, human, organization, network, and domain intelligence.	The four constraints that reflect domain intelligence are determined in stage I. Specific intelligences are also covered by the hypothesis divergence and convergence steps.
The method builds on existing methodologies.	PID-CD as a whole is based on Intelligence Meta-Synthesis, a method used in the field of economic modeling ⁵¹ and association rule mining ¹⁴ . Stage II of PID-CD is based on Shmueli and Koppius' methodology for predictive analytics ^{126, 127} .
The method is efficient with regard to the number of predictive models that are developed.	Throughout stage II of PID-CD, the domain constraints are used to determine which predictive models are not actionable, and need not be developed further.

Table 2.1

Prediction instrument development method requirements as covered by prediction instrument development for complex domains (PID-CD).

PID-CD however, this aspect of goal definition is part of the *domain* constraint definition, and determined later on in the process (but before predictive models are developed). Therefore, what needs to be determined in this stage is just the goal variable, i.e. ‘what needs predicting’.

As mentioned in Section 1.2, the approach to use for prediction depends on the complexity of the domain the goal variable is part of. For a *simple* domain, it is not necessary to use PID-CD, as a literature search for previous occurrences of the same prediction problem would suffice. To help determine whether a domain is complex, we use the definition from Jackson and Keys mentioned earlier [59] (see Section 1.2). This definition is translated to the assessment tool in Appendix J. For domains that score higher than 3 out of 5 on this complexity score, the use of PID-CD is advisable, as reuse of existing solutions is only applicable to the simplest domains, according to the theory of Checkland and Forrester [22, 41].

Another consideration when developing a prediction instrument is the business case for the predictive model. In other words, it needs to be considered how much can be invested in the prediction instrument development, so its predictions still provide a net profit. In practice, if there is little time or money available to develop an instrument, the number of experts or the number of iterations can be reduced. A further discussion of the economic side of prediction instrument development goes beyond the scope of this thesis.

Expert search & selection In PID-CD, domain experts are first consulted to gather the six types of intelligence (see Section 1.2.4), and later to judge the usability of developed predictive models. In this context, an expert is someone who has extensive knowledge of the day-to-day practice, or of hypotheses related to the domain. In other words, an expert is someone who can (or thinks they can) make predictions about the domain. Whether their expertise is sufficient will be established based on the predictive power of the predictive models based on the domain knowledge they contribute.

The process of expert search starts with determining what organizational processes are related to the goal variable of the prediction instrument. Next, a stakeholder analysis is conducted to find all actors involved in these processes. Preferably, each group of actors (primary, secondary, tertiary etc. stakeholders) should be represented in the expert pool. If there are multiple organizations involved, then each actor group should be represented for each organization.

§2.2.2 Stage I: Qualitative assumptions

Stage I(a): Hypothesis divergence. As mentioned before, stage I of PID-CD is about first developing many candidate hypotheses (hypothesis divergence) and then merging them (hypothesis convergence). Hypothesis divergence, which is done by the experts previously selected, should result in many candidate hypotheses, of varying quality. In later steps, the low quality hypotheses are filtered out, but it is important that no potential hypothesis is overlooked in this stage. Gu and Jang [51] suggest to do hypothesis divergence by bringing experts together physically and have them brainstorm on hypotheses that either explain/predict the goal variable or prescribe how it should be predicted.

As complex systems by the definitions of Checkland [22] and Jackson and Keys [59] are about people trying to work together, it is likely that the behavior of these people affect the goal variable in a complex system. Previous research [132] suggests that this behavior is not necessarily to the benefit of the organization, but could in fact be in conflict. If these conflicting interests affect the goal variable, it is unlikely that these interests would be brought forward in a brainstorm session, as it might damage either the reputation of those involved, or the relation between them. Therefore, anonymous brainstorming, where the participants cannot see who suggested which particular hypothesis, is preferable in complex systems. Brainstorm tools such as Spilter [129] support this type of brainstorming.

An alternative to brainstorming is a field study. In such a setup, a researcher would study the experts while they perform their respective tasks, and question them about matters that might influence the goal variable. Each of the hypotheses brought up would then be noted by the researcher, so they can be used in the same way as hypotheses developed through brainstorming. The field study should mix direct observation (*what* are people executing the process doing) with informal interviews (*why* are they doing what they are doing). Pettigrew [112, p.277] provides these requirements for field researchers: “asking and interpreting apposite questions, listening, being adaptive and flexible, being knowingly unbiased, and having a firm grasp of the issues being studied. In addition to these requirements the long distance field worker needs the social and political skills to develop and maintain credibility with a wide range of respondents from different levels and functions inside and outside the focal organisation” [112, p.12] .

As suggested by Pettigrew [112], the researcher’s notes from observation and interviews could be analyzed further by using the iterative techniques from Glaser and Strauss grounded theory [20, 46].

Another source of hypotheses is existing literature. Although literature alone cannot capture the complexities of the domain, theory about similar prediction problems could be useful: this theory could either be used to understand what ‘worked’ in other cases, or as a baseline. Relevant theory can be captured from literature using a *structured review*, such as discussed by Webster and Watson [155].

To ensure that the papers found in the review are of good quality, only those publications (journals and/or conferences) should be searched that are considered to be relevant. To ensure that all relevant papers (and therefore hypotheses) are found, the literature search should contain *forward* (citing papers) and *backward* (cited papers) search.

Stage I(b): Hypothesis convergence. As it is not feasible or economically desirable to create predictive models for every hypothesis developed, the number of hypotheses from the divergence stage has to be reduced. The key to this reduction is to compare the hypotheses and determine which ones are in (full or partial) agreement and which are in conflict. Only those hypotheses that differ significantly need to be tested in the predictive model development stage. In stage II of PID-CD, the emphasis is on the constructs from the converged hypotheses, as these are formed into a data selection strategy. Propositions from the converged hypotheses could be formed into a data cleaning strategy.

Müller provides a framework for classifying inter-hypothesis relations [97]. The scope of this framework is *causal hypotheses*, which covers descriptive hypotheses, explanatory hypotheses, predictive hypotheses and explanatory-predictive hypotheses. Prescriptive hypotheses (i.e. hypotheses for design and action) are explicitly not covered. Every prescriptive hypothesis produced in stage I(a) will therefore be tested in stage II.

The relationship between two hypotheses is classified in one of four mutually exclusive categories: they are *unrelated*, *connected*, *agreeing* or *disagreeing*. To determine the category, first the individual propositions that make up the hypothesis are studied. Propositions describe the relation between constructs, and are either ontological (e.g. A “*is-a*” B or C “*is synonymous with*” D) or causal (e.g. A “*increases*” B, A “*decreases*” B or A “*has no influence on*” B). Diagrams for two example hypotheses, showing all constructs and how they are related (following the diagram standard used in [97]) are shown in Figure 2.1.

Standardization is the next step in classifying the relationship between two hypotheses. This means that synonymous constructs in either hypothesis are treated as identical. In the example in Figure 2.1a, this applies to the

Figure 2.1

Two example hypothesis diagrams. A '=' sign denotes that two constructs are synonymous. A construct nested in another construct means they have a hierarchical 'is-a' relation. A '+', '-' and '0' denote one construct increases, decreases or has no influence on the other construct, respectively

constructs *A* and *B*. Next comes *specialization*, in which more general constructs (e.g. *P* in Figure 2.1a) are replaced by their subconstructs (e.g. *Q* in Figure 2.1a). Note that the hypothesis in Figure 2.1b cannot be standardized or specialized.

If the two hypotheses have zero common constructs after standardization and specialization, they are *unrelated*. If they have exactly one common construct, they are *connected*. If they have two or more common constructs, their relationship is determined by the *total causal influence* between those shared constructs.

To determine the total causal influence, all paths between each ordered pair of common constructs are constructed for both hypotheses. For each path between an ordered pair, the path sign is calculated by multiplying the signs of the propositions on that path using the table in Figure 2.2a. Then, the sum of the signs of each path between the pair of constructs is determined using the table in Figure 2.2b. This is the total causal influence of that pair of constructs.

If the total causal influence for each pair of common constructs is the same for both hypotheses, they are *agreeing*. Otherwise, they are *contradicting*. Looking at the hypotheses in Figure 2.1, there are two common constructs after standardization and specialization, *B* and *Q*. As both constructs are part of a single proposition (i.e. they are directly connected) in both hypotheses, the total causal influence is the sign of that proposition. Both total causal influences are +, so these two hypotheses are *agreeing*.

$\otimes$	+	-	o	?
+	+	-	o	?
-	-	+	o	?
o	o	o	o	o
?	?	?	o	?

$\oplus$	+	-	o	?
+	+	?	+	?
-	?	-	-	?
o	+	-	o	?
?	?	?	?	?

(a)

Calculating the path sign. To calculate path sign, multiply the signs for each path using the table. Adapted from [97, p.4911].

(b)

Calculating total causal influence. To calculate total causal influence, sum the path signs of all directed paths using the table. Adapted from [97, p.4911].

Figure 2.2

Calculation tables for path sign and total causal influence.

To determine how to merge the developed hypotheses, the key is looking at the relation with the goal variable. As this construct should be part of each standardized and specialized hypothesis from the brainstorm/divergence phase, all hypotheses are either connected, agreeing or disagreeing.

Müller's framework and the notion above lead to the following steps for merging hypotheses into one set of hypotheses T for further testing:

- 1.) Translate the hypotheses from the divergence phase to diagrams like those shown in Figure 2.1, i.e. make the constructs and propositions explicit.
- 2.) Standardize and specialize the constructs.
- 3.) For each construct that is part of multiple hypotheses, determine its total causal influence on the goal variable. Group the hypotheses by total causal influence of that construct. If a group contains one hypothesis, add that hypothesis to T . Else, add the hypothesis that has the most constructs (i.e. the hypothesis that is most specific). No duplicates are allowed in T . An example of hypothesis selection for T is shown in Figure 2.3

It is important to note that the propositions and total causal influence in these hypotheses are only used as a method of grouping hypotheses and selecting which hypotheses to test. In stage II of PID-CD, it is primarily the constructs that are considered when developing predictive models. There are exceptions, such as propositions with a 'o'-sign, as these indicate that a construct should be explicitly left out of the predictive model, or *interaction* propositions, i.e. propositions that make another proposition weaker or stronger (moderators

Figure 2.3

Three standardized and specialized example hypotheses that contain the constructs weather (W) and arrival time (A). Hypothesis I and II belong to the same group, as the total causal influence of $W \rightarrow A$ is the same in both hypotheses (+). Hypothesis II has more constructs, and is placed in the set of hypotheses to be tested T . Hypothesis III is also placed in T , as it is the only one in its group.

or quasi-moderators) [97]. If an interaction proposition directly reduces the influence a construct has on the goal variable, a cleaning strategy could be to reduce the effect of that interaction. Other propositions might also become part of a data selection and cleaning strategy – it is the analyst that develops the predictive models who has to decide how to translate hypotheses to data selection and cleaning strategies (see Stage II(a)).

Stage I(c): Constraint definition. There are four constraints that determine which predictive models from stage II are actionable: the data, domain, deployment, and interestingness constraints [14] (see Section 1.2.4). A similar process as with the hypotheses can be applied to determine the constraints, i.e. first brainstorming to propose constraints and then having the experts discuss and select from the proposals.

To determine values for the interestingness constraint, which represents the predictive power of the model, studies with a similar problem can be used as a benchmark.

§2.2.3 Stage II: Predictive modeling

After hypothesis convergence, PID-CD continues with creating predictive models based on the hypotheses in the converged set T . This step in the process is not conducted by the domain experts from the previous stages, but by a technical analyst/data mining expert, i.e. the predictive model developer. Stage II represents a quantitative validation of the hypotheses from stage I.

This stage of PID-CD largely corresponds with Shmueli and Koppius' method for predictive model development [127], but excludes the *goal definition*, *data collection and study design* and *model use and reporting* steps. The first two are covered by stage I, the latter by stage III of PID-CD. *Data preparation and exploratory data analysis* are stage II(a) of PID-CD. *Choice of potential methods* is stage II(c). Finally, stage II(d) and stage II(e) reflect *evaluation, validation and model selection*.

Stage II(a): Translation to data selection & cleaning strategies. There are two activities that are performed for every hypothesis in T in this step. In the first, *hypothesis translation*, the constructs and propositions are translated to a data selection. In other words, the constructs are mapped to variables that are available in the data. This could mean that a construct is substituted by a related variable, if the construct is not directly available. Availability is defined by the constraints determined in stage I(c). These variables are then selected to be part of a *training dataset*. If the total causal influence of a construct on the goal variable is zero, it is not selected for the training dataset.

The variable selection is one part of the translation from hypothesis to data selection and cleaning strategy. If we consider the dataset to be a matrix, these variables represent the columns. The cleaning strategy applies to the rows, i.e. which values for a variable are usable in the predictive model. An example cleaning strategy is to drop rows with missing values or to trim outliers. What strategy to choose could be dictated by the hypothesis if it is a prescriptive hypothesis, or by any moderating interaction propositions in the hypothesis. Else, it depends on the insight and analysis of the predictive model developer.

This insight could be gathered from an exploratory data analysis (EDA) [126, 127]. Exploratory data analysis “consists of summarizing the data numerically and graphically, reducing their dimension, and ‘preparing’ for the more formal modeling step” [126, p.297]. In other words, its goal is to better understand the available data and how it relates to the *hypothesis* that is tested. It might well be that EDA shows that the hypothesis has low *explanatory* power (i.e. weak statistical relations between independent constructs and goal variable). Even so, that does not mean the final model has low *predictive* power.

Stage II(b): Reduction by data & domain constraints. Together, the data selection and cleaning strategies form the set D . The next step is to further reduce this set based on the *data* and *domain* constraints. The data constraint means a strategy is discarded if it requires data (i.e. a variable) that is not available, or if the data cleaning strategy would not leave enough data to train a predictive model. The domain constraint means a strategy is discarded if the data required by the strategy would violate business rules (like the use of business-sensitive data) or would violate ethical rules (like the use of data that can be traced back to individuals).

Stage II(c): Technique & parameter selection. Shmueli and Koppius recommend the use of ensemble methods for training predictive models [126, 127]. These methods combine the results from multiple (weaker) models to make one compound predictive model, e.g. by averaging the predictions of the models it combines. Two examples of ensemble methods that are known to create powerful predictive models [132, 134] are Breiman’s *Random Forests* [11] and Freund and Schapire’s *AdaBoost* [43, 44].

To reduce the risk that technique-specific issues cause poor performing models, we propose the use of a sample of each category of classification/regression techniques. Besides ensemble methods, these categories [142] include clustering techniques, such as *k-Nearest Neighbors* or *Support Vector Machines* [32], and decision tree techniques like *Classification and Regression Trees* (CART) [12]. *Neural networks* can also be used for regression and classification.

Besides the selection of techniques, the parameters to that technique influence the performance of the produced predictive models. Most of these parameters are technique-specific, but for example the size of the training set is a general parameter. Techniques that are sensitive to overfitting or overtraining (e.g. decision trees and neural networks) work best with smaller datasets³. Others, like *Random Forests*, do not suffer from overfitting or overtraining [11], and can therefore safely be used with large training sets. The best choice for technique parameters depends on the expertise of the analyst/predictive modeler.

³An detailed explanation of overfitting/overtraining is beyond the scope of this thesis, but can be found in [142]. In short, a predictive model is overfitted if it is too well adapted to the training set and its particular distributions. As the training set is not necessary representative of the test data, this harms predictive power.

Stage II(d): Model training. All strategies that have not been discarded by stage II(b) are combined with each technique from stage II(c) to train predictive models. To train a model, the predictive modeler first creates a cleaned dataset using the data selection and cleaning strategy. This dataset is subsequently split into two non-overlapping parts: a training set that is used by the technique to create the predictive model, and a test set that is used to determine the performance of that model. The produced model takes a set of independent variables (constructs) as input and produces a prediction of the goal (dependent) variable as output.

As mentioned before, the prediction goal (e.g. predicting a categorical or numerical outcome) affects what models to develop and what techniques to use [127]. What types of outcome are actionable is determined by the domain constraint, and the predictive modeler should only develop models that fit with that constraint.

Stage II(e): Reduction by interestingness, deployment & domain constraints. The final step of stage II consists of an evaluation of the developed predictive models. As presented in Section 1.2.4, the interestingness constraint defines both the performance metrics and the acceptable range of values for that metric. An example interestingness constraint is that precision and recall⁴ are at least 50 and 70 percent, respectively. The deployment constraint specifies for example the infrastructure the model has to interact with (what the inputs and outputs are) and the speed at which the model can make predictions, or the speed at which a new predictive model could be made using the same technique. Especially ensemble models and neural networks could be time consuming to train, depending on their parameters. Finally, the domain constraint specifies any other conditions that make a predictive model actionable.

⁴These two metrics are based on the number of *true positives*, *false positives*, *false negatives*, and *true negatives*, and are calculated for each individual class. For an example class *A*, these are determined as follows: if the predicted class and *true* class are both *A*, that is a *true positive* (*tp*); if the predicted class is *A*, but the true class is not, that is a *false positive* (*fp*); if the predicted class is not *A*, but the true class is *A*, that is a *false negative* (*fn*); and if neither the predicted class nor the true class is *A*, that is a *true negative* (*tn*). Given these four values, the two metrics are calculated as follows:

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

Figure 2.4

An overview of PID-CD for predictive model development. The main process of PID-CD starts with *Goal definition* and ends in *Model convergence*. *Constraint definition* can be executed in parallel with *Hypothesis divergence/convergence*, but has to be finished before the first *Reduction* step. *Technique selection* is the responsibility of the predictive model developer and as such, it can be done entirely separate from the rest of process.

§2.2.4 Stage III: Model convergence

The final stage of PID-CD represents a qualitative validation of the developed predictive models. Only those models that remained after stage II(e) are up for validation in this stage. The goal of stage III is to select one predictive model, which will be the basis of the prediction instrument.

The decision makers from the domain (e.g. managers) decide which predictive model to use. If there is more than one decision maker, or if the decision makers do not directly agree, that decision is made using either *voting*, or a group decision framework such as the *Analytic Hierarchy Process* (AHP) [121]. If there is a predictive model that fits with the decision maker's demands, then the data selection and cleaning strategy, technique, technique parameters, and constraints that were used to create that model form the *prediction instrument*.

Alternatively, if there is either no predictive model that fits with the demands, or there was no predictive model remaining after stage II, parts of the method should be revisited. Possibly other experts should be selected, so hypothesis divergence and convergence leads to different hypotheses, or the constraints should be revised, so they are more inclusive. Then, stage II and stage III are re-executed. This should only be repeated until saturation is reached, i.e. when the predictive models no longer improve with regard to the decision makers' demands.

§2.3 Prediction instrument development with soft-exclusive domain analysis

To be able to compare the results (i.e. predictive power) from a prediction instrument developed with PID-CD (i.e. with soft-inclusive domain analysis), this section presents a method for prediction instrument development with a soft-exclusive domain analysis. The method, *prediction instrument development with soft-exclusive domain analysis* (PID-SEDA), follows the same basic outline as PID-CD, so the main difference is the domain analysis, and the rest of prediction instrument development is comparable. So, stage I of PID-SEDA collects hypotheses from a literature review, stage II creates predictive models, and stage III selects one predictive model and turns it into a prediction instrument. Stage I is based on Webster and Watson's method for literature review [155], and like PID-CD, stage II is based on Shmueli and Koppius' method for predictive analytics [126, 127].

§2.3.1 Stage I: Assumptions

The first stage of PID-SEDA starts with determining the type of domain, and what abstract domain this is an instance of. For example, ‘predicting truck arrival time at distribution center XX’ could be considered an instance of the ‘predicting truck arrival time’ domain. The (abstract) domain is the starting point for a literature review. The goal of this review is to gain domain understanding, which should be translatable to a data selection and cleaning strategy. The review process starts with a *query*, next *abstracts* are reviewed, then *full papers* are reviewed, and finally, one or more hypotheses are extracted. The process is described in detail below.

Query The starting point for the literature review is the query. It should be tuned so it captures all relevant papers, but for efficiency’s sake, does not give an excess amount of irrelevant results. For the latter reason, the choice of search engine is also relevant, as it must be able to properly filter results that are de facto irrelevant. The query should contain the name of the domain, including synonyms, as well as synonyms of ‘variable’ and ‘prediction’. This ensures that the results discuss prediction of the domain, and describe what variables are used. Finally, to maximize the relevance of the query results, it makes sense to select which publications (journals or conferences) are relevant. Here, tools like the ISI-JCR [144] or the AIS Senior Scholar’s Basket of Journals [5] as well as scientometric papers on academic relevance are useful. Note that which sources should be used depends on the domain.

Abstract review In this phase, the abstracts of the resulting papers are reviewed. The goal of the phase is to determine which of the paper are highly relevant, so their full text can be studied. A list of abstract inclusion or exclusion criteria is used to distinguish irrelevant from highly relevant papers. Example inclusion criteria are ‘discusses the domain’ and ‘discusses prediction’, and an exclusion criterion could be ‘does not use empirical data’.

Full paper review The full paper review follows the same pattern of the abstract review, but with more strict inclusion and exclusion criteria. Here, only the papers that discuss a predictive model of the required domain in sufficient detail would be considered relevant enough. That means that only papers that make clear what variables are used and what data is used would remain after this stage.

Forward & backward search Forward and backward search ensures that potentially relevant papers are not missed due to missing publications or terms in the query, or any other reason. In forward search, all papers citing the highly relevant papers remaining after the full paper review are reviewed using the same process and criteria as before (first abstracts, then full papers are reviewed). The same is done for the papers cited in the highly relevant papers. If desirable, the forward and backward search could be repeated until saturation is reached.

Paper convergence The final step of stage I of PID-SEDA is translating the findings from the complete set of relevant papers into a data selection and cleaning strategy. The basis for selecting variables is the frequency with which they are mentioned in the literature, although variables that are only mentioned in a minority of papers could also be considered, if there is strong evidence of their influence on the goal variable.

We consider two approaches to converging the set of papers into testable hypotheses. The first is to consider each paper's set of variables a hypothesis, and then apply the *hypothesis convergence* from PID-CD (see Section 2.2.2) to determine which hypotheses to test in stage II. Alternatively, the most frequently occurring variables could form the hypothesis. The latter is more in line with the soft-exclusive notion of *abstract domains*, which have a minimal set of attributes that are shared by each instance (see Figure 1.2). Similar to the discussion in Section 2.2.2, papers that represent prescriptive hypotheses (hypotheses for design and action) cannot be converged using these techniques, and will all have to be tested in stage II.

§2.3.2 Stage II: Predictive modeling

The second stage of PID-SEDA is an analogue for stage II of PID-CD (see Section 2.2.3), with some exceptions. As stage I of PID-SEDA does not result in a full set of constraints (see Section 1.2.4), there is no method for discarding models that would likely be not actionable. The only constraint for PID-SEDA is the data constraint, as models cannot be developed without data. The steps in stage II are as follows: first, hypotheses from the papers are translated to data selection and cleaning strategies. If there is enough data, the next step is to create a predictive model, using a previously selected technique. The predictive modeling and technique selection steps are exactly the same as those from PID-CD (discussed in Section 2.2.3), and will therefore not be repeated here.

§2.3. Prediction instrument development with soft-exclusive domain analysis

Translation to data selection & cleaning strategies If the papers discuss data cleaning strategies, these are combined with the variable hypotheses to form data selection and strategies. However, as the domains and data discussed in the papers is likely to be at least somewhat different from the present domain and its data, it is likely that the discussed data cleaning strategies cannot be used directly. The same holds for prescriptive hypotheses from the literature. In those cases, developing a cleaning method relies on the expertise of the analyst.

Reduction by data constraint After one or more data selection and cleaning strategies are developed, those that are not supported by data are eliminated. Each remaining strategy is converted to a predictive model, following the same method as described in Section 2.2.3.

§2.3.3 Stage III: Model convergence

In the third and final stage of PID-SEDA, one predictive model is selected to be the basis for a prediction instrument (for the distinction, see Section 1.3). As PID-SEDA does not explicitly gather experts from the domain to decide which predictive model is best, the choice of predictive model is based on predictive power. Regardless of other concerns, the predictive model that scores highest on the metric chosen by the analyst, or potentially that suggested by literature, will be the basis for a prediction instrument.

3 Soft-exclusive domain analysis for a simple domain

In this chapter, we present a case study for prediction instrument development in a simple domain. The goal of this chapter is to demonstrate the predictive power of a prediction instrument developed in a simple domain with soft-exclusive domain analysis. The method used for prediction instrument development is PID-SEDA, as described in Section 2.3. As mentioned in Chapter 1, it is unlikely to find a real world domain that is not affected by behavior, culture, politics, or other soft factors. That also applies to the case described in this chapter: it is a simulation of a container terminal for the purpose of benchmarking predictive performance. Benchmarking is the main area where we theorize that soft-exclusive domain analysis is a good fit for prediction instrument development.

When developing a predictive model, various metrics are used to evaluate its predictive power [126, 127]. Predictive model development is an iterative process, where the models are improved until *saturation* is reached, i.e. until the metrics no longer improve with changes to the model. As there is often no theoretical ‘best’ or even ‘good’ performance known for the domain, it is difficult to assess how good the final predictive model *is* with regard to how good it *could* be. We resolve the issue of comparing predictive model performance by developing a benchmark predictive model. Note that as mentioned in Chapter 2, it is not necessarily feasible from a business perspective to keep improving the predictive models: the cost of predictive modeling could outweigh the potential gains.

The domain we develop the benchmark for is *predicting truck turnaround time* at a container terminal. Truck turnaround time is the total time a truck spends between entering and exiting a terminal. In that time, it could either

drop off a container, pick up a container, or do both. Transport companies in the Port of Rotterdam often make multiple terminal visits per day. They pick up containers, deliver their contents to hinterland depots, and then change the empty container for a full one at the terminal. For these companies, accurately predicting turnaround time means that they can plan the order of terminal visits to minimize total turnaround time. This could make the difference between making four and five trips a day, considerably affecting their bottom line. Therefore, turnaround time of a truck is an important measure for trucking companies, but we could not find existing models in literature that predict turnaround time. Hence there is a need for a benchmark predictive model.

This chapter is organized as follows. First, we present a description of the simulated terminal. We use a simulated terminal so we can control the amount of noise in the data. Next, we describe the use of PID-SEDA (see Section 2.3) to create a benchmark prediction instrument. As the data for this prediction instrument is simulated, stage I of PID-SEDA takes a somewhat different form: domain analysis is not used to gather assumptions about what affects the prediction outcome, but to create a realistic terminal simulation. As the simulation's goal is to produce noise-free data in sufficient quantity, which is usually an element of the data constraint, that constraint definition is also discussed as part of terminal construction. In Stage II, there are no cleaning strategies, as the data is noise-free by construction, but otherwise it follows the PID-SEDA method, as does stage III.

§3.1 Case description, assumptions, and simulation construction

The benchmark prediction instrument developed in this chapter predicts turnaround time at a container terminal. Turnaround time is the total time between a truck entering the terminal and exiting the terminal. In that time, a series of tasks has to be completed. First, the trucker has to register before being allowed onto the container area of the terminal. Next, a container is unloaded and/or loaded, and the trucker exits the container area after getting the appropriate permissions.

In real world container terminals, the time it takes to complete each of these activities depends on behavior, culture, politics and other soft factors. For example, truckers are not always motivated to work quickly, and crane operators may take notoriously long breaks (see Chapter 5 for an analysis of a real container terminal). This case study sets out to have those activity durations follow the distributions prescribed by theory.

Figure 3.1

Overview of the simulated terminal. Trucks arrive at GATE IN, where they are let on to the terminal after being processed. Next, they either go to the UNLOADING or LOADING crane. If the truck first unloads, it can later go to the LOADING crane. Finally, trucks head to the GATE OUT, and leave the terminal after being processed again.

The reason that predicting turnaround time is interesting for transport companies is that it allows them to be as efficient as possible. Often, trucks travel back and forth between container terminals in the port and hinterland depots, making multiple trips in one day. If the transport company can predict the turnaround time at each of the terminals that need to be visited in a day, it can order visits in such a way as to minimize the total turnaround time. This could enable each trucker to make one or more extra visits per day, significantly affecting the company's "bottom line".

To determine how good those turnaround predictions for different terminals are, is what the benchmark predictive model is used for. As it is not affected by terminal-specific noise, it is comparable to each terminal, regardless of the behavioral, cultural, and political factors at play at that terminal and between the terminal and transport companies.

Figure 3.1 shows the basic structure of our simulation, the layout of a simple terminal. The layout is based on actual terminals in the port of Rotterdam in the Netherlands. Each truck arriving at the terminal has to go through the GATE IN, before any empty containers are removed at UNLOADING by a yard crane (YC) and a new container is put on the truck by another YC at LOADING. The figure also shows the distances between points on the terminal, and the different routes a truck can take depending on its orders, i.e. loading, unloading, or both loading and unloading a container.

§3.1.1 Simulated terminal

The variables related to truck turnaround time were taken from the papers selected through a structured literature review. The review methodology and the details of the findings from the review are described in Appendix I. These are used to create the simulated terminal and to select variables for the predictive model.

The variables from the literature review are represented in the simulated terminal as follows: congestion, demand, arriving vessels and time of day are represented by *truck arrival distribution*; capacity and amount of equipment, operator style and error, job duration, gate processing rate, stack configuration, and storage sequence are represented by *crane and gate processing rate*; the number of containers to (un)load and travel distance are represented by the *route through the terminal*. Weather and refrigerated containers (reefers) were not implemented in the simulation.

There are three types of queues in the simulated terminal: a *road queue*, representing a stretch of road (e.g. the 425 meter stretch of road in the top middle in Figure 3.1); a *crane queue* (e.g. UNLOADING), representing a crane

and the waiting line in front of that queue; and a *gate queue*, representing the truck processing (and waiting line) at the entry and exit of the terminal (e.g. GATE OUT).

Queues The terminal is modeled as a network of queues Q , as suggested by [1, 27]. Each $q \in Q$ has a set of previous queues $Q_{pre} \subset Q, \forall q' \in Q_{pre} q \neq q'$. For GATE IN, Q_{pre} is the empty set $\{\emptyset\}$. Furthermore, each queue has a set of trucks q_T , which represents queued trucks. Trucks are processed in the order they arrive (FIFO).

Queue rate Each queue has a rate r_q , which determines, combined with the time needed to traverse the queue's waiting area, $\min(t_q)$ (see below), how long it will take to process a single truck. This rate is set for each individual truck, following a queue-specific distribution. The selected papers from the literature review suggest the following distributions for gate processing rate and YC processing rate (based on empirical data):

$$\begin{aligned} r_{crane} &\sim \text{Triangular}(\text{left} = 1.2, \text{mode} = 2.0, \text{right} = 3.4) [109-111] \\ r_{gate} &\sim \text{Erlang}(\mu = 2.44, k = 4) \equiv \Gamma(\theta = 0.61, k = 4) [26] \\ r_{road} &= 0 \text{ (The waiting area length determines the queueing time)} \end{aligned}$$

Minimum time at queue Trucks are modeled to have a length l_T of twelve meters. Each queue has a length $l_{q,q \in Q}$ in meters, which determines the maximum number of trucks that can be queued:

$$q_{max} = \lfloor \frac{l_q}{l_T} \rfloor$$

The maximum speed of trucks on our terminal is set at $V_{max} = 30 \text{ km/h} = 8.34 \text{ m/s}$, and trucks are assumed to have an acceleration $a_T = 2 \text{ m/s}^2$ and deceleration $d_T = 2.5 \text{ m/s}^2$.

Combined with the queue length l_q , this gives the minimum time to traverse the length of a queue (i.e., a stretch of road). Consider a truck that starts the queue at t_0 with speed zero, accelerates until it reaches the maximum speed, and then decelerates to arrive at the end of the queue at t_{end} with speed zero. The point in time where it reaches V_{max} is labeled B, the point where it has to start decelerating to stop in time is labeled C. This is shown in Figure 3.2.

The time between t_{start} and t_{end} is the minimum queuing time $\min(t_q)$. It is the minimum, as this only occurs when there is just one truck in the queue, otherwise, it would have to wait for other trucks to clear the queue. The minimum queuing time is calculated as follows:

$$\begin{aligned}
 B &= \frac{V_{max}}{a_T} \\
 s &= \text{Distance covered} \\
 s(t_0 \rightarrow B) &= \frac{1}{2} * a_T * B^2 \\
 s(C \rightarrow t_{end}) &= \frac{1}{2} * d_T * \left(\frac{V_{max}}{d_T}\right)^2 \\
 \text{if } s(t_0 \rightarrow B) + s(C \rightarrow t_{end}) &< l_q : \\
 s(B \rightarrow C) &= l_q - (s(t_0 \rightarrow B) + s(C \rightarrow t_{end})) \\
 C &= B + \frac{s(B \rightarrow C)}{V_{max}} \\
 t_{end} = \min(t_q) &= C + \frac{V_{max}}{d_T} \\
 \text{if } s(t_0 \rightarrow B) + s(C \rightarrow t_{end}) &\geq l_q : \\
 h &= \text{Maximum speed to reach} \\
 h &= \sqrt{\frac{2 * l_q}{\frac{1}{a_T} + \frac{1}{d_T}}} \\
 t_{end} &= \text{Time needed to reach } h \\
 &+ \text{Time needed to reach zero from } h \\
 t_{end} = \min(t_q) &= \frac{h}{a_T} + \frac{h}{d_T}
 \end{aligned}$$

Arrival at gate The selected papers from the literature review suggest four different distributions for the number of trucks arriving at GATE IN: either a β distribution [25], an exponential distribution [26], a uniform distribution [25, 101], or a Poisson distribution [167].

The Poisson and β distribution are both used in a context where either trucks arrive to pick up containers within a certain interval/window after a ship has arrived at the terminal, or the trucks have an appointment [25, 167]. Chen *et al.* and Namboothiri and Erera suggest that the arrival of trucks

Figure 3.2

Speed of a truck on a queue over time. B is the point in time at which the truck reaches its maximum speed. C is the point in time when the truck has to start decelerating to be at a standstill at the end of the queue. Given a truck's acceleration and deceleration, and the length of the queue (distance), the total time to cover the queue can be calculated as shown by the formulas on page 63.

throughout the day has multiple peaks, such as early morning and early afternoon [26, 101]. So, literature suggests two variants for truck arrivals with four distributions: ship-arrival based ‘windows’ with a β or Poisson distribution and ‘peaks’ with a uniform or exponential distribution.

Chen *et al.* are most specific about truck distribution, as they also give the parameters of the β distribution: $\beta(\alpha = 1.25, \beta = 3.25)$ [25]. Therefore, we have implemented the ‘window’ variant with a β distribution. For the ‘peak’ variant, Namboothiri and Erera (2008) provide specifics for truck arrival distribution [101], whereas Chen *et al.* [26] only mention that truck arrival is distributed exponentially. Namboothiri and Erera(2008) describe three scenarios: uniform, where the same number of trucks arrive throughout the day; midday-heavy, where three times more trucks arrive around the middle of the day than at the start and end; and morning-afternoon-heavy, where three times more trucks arrive at the start and end of the day than at the middle of the day. Note that their research is about capacity requested by a trucking company, rather than actual arrivals, but we assume that these would follow a similar distribution. Of these distributions, the ‘window’ variant and the morning-afternoon-heavy variant are developed further. In the remainder of this chapter, the first is referred to as a β distribution, the second as a High-Low-High (HLH) distribution. Figures 3.3a and 3.3b show both implemented distributions.

Pseudocode Algorithm 1 shows pseudocode for the main element of our simulation, which was implemented in Python. By using a well known open source programming language such as Python, we ensure that our simulation can be easily used and extended by others, which might not be the case with a tailored simulation program. This algorithm is run once every second of the duration. Line 4 refers to a function that generates a truck visit with a random selection of one of the following three ‘orders’: it can unload an empty container and then load a full one, or only either unload or load. In line 5, the queue checks if the current truck has been in the queue longer than the queue service time s_q , which is the inverse of the queue rate r_q (i.e. $s_q = \frac{1}{r_q}$). That rate is drawn from the applicable distribution at the moment the truck enters the queue. Lines 9 and 11 show the mechanism for dealing with full queues. If the the next queue (a road, crane, or gate) for a truck is completely full, it has to stay at the queue where it is and that queue stops processing. As soon as a queue manages to send a truck to a next queue (or the end of the

(a)

B(1.29, 3.25) probability density function for truck arrivals, with 10 'windows' of 144 minutes

(b)

Morning-afternoon-heavy probability density function for truck arrivals, with three times more trucks at the start and end of the day than at midday

Figure 3.3

Probability density functions for truck arrivals at the terminal. The total black area in each plot represents the sum of all trucks arriving over the course of the day, which is the same for plot (a) and (b). Therefore, it should be noted that the actual values for P are higher for (b) than for (a). However, for simplicity, we have excluded the values for the y -axis here.

```

1  $t \leftarrow 0$ ;
2 for  $t \leftarrow 0$  to simulation duration do
3    $N_t \sim$  truck arrival distribution;
4   Generate  $N_t$  trucks;
5   foreach queue in queues do
6     if queue is running and  $|queue| \neq 0$  and  $t_{queue[0]} > s_{queue}$  then
7       if next queue is not full then
8         Move first truck in queue to next queue;
9         Notify  $Q_{pre}$ ;
10      else
11        Stop queue;
12      end
13    end
14  end
15 end

```

Algorithm 1: Pseudocode for the simulation

terminal), the queue notifies all previous queues that they can start processing again. Every time a truck leaves the terminal, its start time, end time, and orders are recorded in a dataset.

§3.2 Stage I: Assumptions

§3.2.1 Goal definition

In Shmueli and Koppius' (2011) method [127], goal description is about determining what the output of the predictive model should be, which determines a.) how to develop a predictive model and b.) how to determine its performance/predictive power. For our benchmark truck turnaround time prediction, the dependent variable of the predictive model is turnaround time: the total time spent between entering and leaving the terminal. There are two options for the type of this dependent variable: continuous or categorical.

In the continuous case, the predictive model is used to exactly determine turnaround time, e.g. '39 minutes'. This is called *regression*. In the categorical case, turnaround time is predicted in intervals, e.g. '35-40 minutes'. This is called *classification*. In this chapter, we will compare both alternatives.

The performance of the predictive model we are developing serves as a benchmark. We therefore record a wide range of metrics, so that the results can be compared to as many other predictive models as possible. For *classification*, the main metric is accuracy: the number of correctly classified instances divided by the total number of instances. As we run multiple tests of each predictive model, we also record the distribution of accuracy across these tests.

For *regression*, the metrics are all error metrics: they measure the difference between the predicted turnaround time and actual turnaround time for each instance. We record the following metrics for regression: *mean error*, *mean absolute error*, *mean square error*, *root mean square error*, and *mean absolute percentage error*. Given a set of instances with predictions I , the cardinality of I (i.e. the total number of instances) n , a prediction p_i for each instance $i \in I$, and an actual value a_i for each instance $i \in I$, these metrics are calculated as follows:

$$\text{Mean error} = \sum_{i \in I} \frac{p_i - a_i}{n} \quad (3.1)$$

$$\text{Mean absolute error} = \sum_{i \in I} \frac{|p_i - a_i|}{n} \quad (3.2)$$

$$\text{Mean square error} = \sum_{i \in I} \frac{(p_i - a_i)^2}{n} \quad (3.3)$$

$$\text{Root mean square error} = \sqrt{\sum_{i \in I} \frac{(p_i - a_i)^2}{n}} \quad (3.4)$$

$$\text{Mean absolute percentage error} = \frac{1}{n} \sum_{i \in I} \left| \frac{p_i - a_i}{a_i} \right| * 100 \quad (3.5)$$

§3.2.2 Data constraint

The data constraint, which is the only constraint determined for PID-SEDA, determines the minimal acceptable data quality and quantity. As the data used here is drawn from a simulation, its quality is guaranteed by the quality of the simulation. There are two requirements that the data should meet: first, it should be realistic, both in distributions of the variables as in availability of the variables in empirical settings, which is called *ex ante availability* [127]. Second, the data should not contain any noise, such as human noise [134], which is referred to as *measurement quality* [127].

As the simulation is well grounded in theory, the first requirement is met by the datasets. All distributions used in the simulation are based on theory from literature, which should ensure they reflect the real world. The same holds for the number of variables used. The second requirement is met by the nature of the simulation, which inherently does not contain any human noise. Furthermore, the simulation itself is rigorously tested, to ensure no mistakes in the simulation design affect the data quality.

§3.3 Stage II: Predictive modeling

§3.3.1 Data selection strategies

In this section, we present our method for generating data for a benchmark predictive model. We want the data for the benchmark to reflect the actual situation at a container terminal as good as possible. This means that we need to find what aspects of a terminal affect truck turnaround time, for which we use the same structured literature review that was used to develop the simulated terminal. That way, the aspects we use are well grounded in established theory. Next, in Section 3.1.1, we use these aspects to create a simulated container terminal.

We use the simulated terminal from Section 3.1.1 to generate data to create (train) and test a predictive model. The number of datasets we generate is based on two considerations. First, as mentioned in the previous section, there are two variations in the distribution of truck arrivals, meaning that at least two datasets should be tested. Second, as all the selected papers from our review suggest that congestion and the related terminal capacity are relevant, we also generate a congested and an uncongested dataset, bringing the total to four.

To construct an uncongested or congested dataset (i.e. one with ‘high’ and one with ‘low’ traffic), we vary the total number of trucks arriving per given time period. The terminal is congested if one or more queues have a higher input flow than output flow. Given that the least-capacity queues (the in and out gates) can process around 20-25 trucks per hour, this means that the total number of trucks arriving per hour should be greater than 25 at some points in time for the congested dataset, and never greater than 20 for the uncongested dataset.

Each simulation will run for 1 000 hours of simulated time (41 days and 16 hours), in order to make sure that the simulation reaches a ‘congested’ state. During that time, some 15 000 to 30 000 trucks arrive at the terminal, depending on the truck arrival distribution.

The available variables for each terminal visit in the dataset (as per the findings of the literature review) are the following: *start time*, i.e. the time the truck enters the terminal’s first queue (in seconds); *end time*, i.e. the time the truck leaves the terminal’s last queue (in seconds); *load*, whether the truck picks up a container (true or false); *unload*, whether the truck drops off a container (true or false); and *congestion*, the total number of trucks in a queue divided by the total capacity of queues *at the time the truck entered the terminal* (ranging from zero to one).

Data quality Following Shmueli and Koppius’ method [127], this section deals with cleaning data and partitioning the datasets into training and holdout sets. As the data used in this research is from a completely controlled environment (the simulated terminal), it should be free of noise by design. However, for completeness’ sake, we address potential data quality issues below.

In Shmueli and Koppius’ (2011) method, data cleaning and selection revolves around two constraints: measurement quality and *ex ante* availability, i.e. the availability of a variable at the time of prediction [127]. Measurement quality is about the quality of the data used. As stated in the previous section, the data is fully clean, i.e. the generated data is free of noise. The EDA also shows that at least one of the variables in the dataset (congestion) is a good predictor of turnaround time – the relation congestion-turnaround time is significant at the $p < 0.01$ level, with $0.66 \leq R^2 \leq 0.95$, depending on the dataset. The relation of the route and start time variables to turnaround time does not appear to be as strong, but a) they are mentioned in the literature to be important predictors, and b) Shmueli and Koppius suggest that the choice of variables for a predictive model should be wide, “to better allow for the discovery of new relationships” [127, p.564].

With regard to *ex ante availability*, there is one variable that might not be available at the time of prediction, and that is *congestion*. As congestion can only be directly measured by a terminal operator, the availability of this variable strongly depends on the willingness of the terminal operator to supply that data. To take this into account in our benchmark, we will use both a dataset that contains all three variables (route, start time, and congestion) and one that only uses start time and route. Table 3.1 gives an overview of the different datasets that will be used for training predictive models.

<i>Arrival distribution</i>	<i>Congested</i>	<i>Variables</i>	<i>Number</i>
β	Yes	Congestion, route and start time	1
		Route and start time	2
	No	Congestion, route and start time	3
		Route and start time	4
High-low-high	Yes	Congestion, route and start time	5
		Route and start time	6
	No	Congestion, route and start time	7
		Route and start time	8

Table 3.1
Datasets for predictive model development

(a)
 β arrival distribution, uncongested

(b)
 β arrival distribution, congested

(c)
 High-low-high arrival distribution, uncongested

(d)
 High-low-high arrival distribution, congested

Figure 3.4
 Distribution of truck turnaround times, with the best fitting distribution (based on the moment-matching goodness-of-fit statistic).

(a) β arrival distribution, uncongested

(b) β arrival distribution, congested

(c) High-low-high arrival distribution, uncongested

(d) High-low-high arrival distribution, congested

Figure 3.5
 Distribution of turnaround times for trucks that only unloaded a container, only loaded a container, or unloaded and loaded a container.

(a) β arrival distribution, uncongested

(b) β arrival distribution, congested

(c) High-low-high arrival distribution, uncongested

(d) High-low-high arrival distribution, congested

Figure 3.6
Truck turnaround time versus congestion at the terminal for each truck arrival distribution.

(a) β arrival distribution, uncongested

(b) β arrival distribution, congested

(c) High-low-high arrival distribution, uncongested

(d) High-low-high arrival distribution, congested

Figure 3.7
Autocorrelation of truck turnaround times.

§3.3.2 Exploratory data analysis

Figure 3.4 shows the distribution of turnaround time for the four truck arrival distributions, and which statistical distribution fits best. From Figure 3.4a, we find that the uncongested β arrival distribution results in turnaround times that fit a β distribution¹, albeit with different parameters. None of the other turnaround distributions fit a well-known distribution, i.e. a normal, exponential, lognormal, logistic, gamma or uniform distribution. Clearly the four truck arrival distributions lead to quite different turnaround time distributions, in both their shape and the range of turnaround times.

Figure 3.5 also shows distributions of turnaround time, but differentiates between trucks that arrived for only loading, only unloading, or that both unloaded and loaded a container. For all but one arrival distribution, there do not seem to be major differences between loading, unloading or both. The exception is 3.5b, where unloading seems have its peak values at shorter turnaround times than loading or unloading and loading.

Figure 3.6 shows the relation between congestion at the terminal and turnaround time. Here, congestion describes how much of the total capacity of the queues along a truck's route is filled. For example, if there are three queues with capacities (20, 10, 15) and queue lengths of (18, 6, 5) respectively, the congestion is $\frac{18+6+5}{20+10+15} = 0.644$. For each truck arrival distribution, turnaround time goes up as congestion goes up. This relation is significant at the $p < 0.01$ -level, with $R^2 = 0.66$ for the uncongested β arrival distribution, $R^2 = 0.95$ for congested β arrival distribution, and $R^2 = 0.85$ and $R^2 = 0.95$ for the uncongested and congested high-low-high distribution, respectively.

Figure 3.7 shows the relation of a truck's turnaround time to previous trucks. It is clear from the plots that there is a 'seasonality' to turnaround time, i.e. it follows a wave-like function over time. The period of this wave differs for each arrival distribution, as can be seen in Figure 3.7.

§3.3.3 Technique selection

This section describes the data mining techniques used to make a predictive model from the four datasets. As our goal is to create a predictive model that is as good as possible, we will use several different techniques, to determine which performs the best for each dataset.

¹To fit a β distribution, which only has values between 0 and 1, the probability density function is multiplied by $\max(t_{\text{turnaround}})$

Based on what previous research [132, 134] found to be a well-performing data mining algorithm, we use Breiman’s Random Forests [11] and a variation on Breiman *et al.*’s Classification and Regression Trees (CART) [12]². Finally, for regression tests, we also test linear regression, as the EDA has shown significant correlations between the predictors and turnaround time.

§3.3.4 Model training

As we have eight datasets, two types of test (classification and regression), and three techniques (CART, Random Forest, and, for the regression models only, linear regression), we developed a total of forty predictive models (8 datasets * 2 classification techniques + 8 datasets * 3 regression techniques). All predictive models were developed using the *R* suite of statistical software [115], and the *R* implementations of Random Forests [87] and CART [143].

With regard to dataset partitioning, we used a ten-fold cross validation. In this approach, a dataset is split randomly into ten equal size partitions. There are then ten iterations over the data, each time selecting one partition as hold-out and the other nine as training set. After ten iterations, we average the results, with each partition having been used as a holdout set exactly once. This approach to partitioning should eliminate bias/“overoptimistic predictive accuracy” [126]. In this section, we present the performance results of each predictive model.

Classification models Table 3.2 shows the performance for each classifier in each of the sixteen classification tests. Each value indicates how many rows of data were correctly classified as a fraction of the whole test set. Correct implies that the algorithm’s prediction of the 5-minute interval of a truck’s turnaround time matches the actual interval of the truck’s turnaround time.

We find that Random Forest is the better performing of the two algorithms, but not by a big difference. The mean performance for the β arrival distribution datasets is around 0.48, and around 0.35 for the High-low-high arrival distribution datasets. Only using the start time and the route through the terminal has a strong effect on accuracy, reducing it by 30 to 50 percent.

Regression models Table 3.3 shows the performance for each regression technique in each of the twenty-four regression tests. Each value represents a different error statistic, expressed in seconds. Compared to the classification re-

²An explanation of the workings of these algorithms goes beyond the scope of this chapter, and should be considered a textbook matter

sults, there is not one technique that stands out over the others, although linear regression seems to perform better than the other algorithms. This can be explained by the significant relations between the predictors and turnaround time (see Section 3.3.2).

Similar to the classification results, the use of only route and start time as predictors *generally* negatively influences performance. Exceptions are the mean error, which is in fact the worst performing of all, when using only these predictors for the β -datasets.

§3.4 Discussion

In this chapter, we set out to demonstrate how one can create a benchmark predictive model for a particular case. We then created a benchmark based on several performance metrics for predicting turnaround time. The predictive models show that low congestion makes for the best predictions, which is in line with literature. The best accuracy for classification was around 46 percent for the uncongested dataset with a β arrival distribution. The best regression models have a mean absolute error of around 195 seconds, again for the uncongested dataset with a β arrival distribution. It would seem that the β -distribution leads to the most predictable results.

In our simulation we avoid factors such as behavior, politics or other human factors [3] that can create noise and significantly affect the outcome of a prediction [132, 134]. These potentially noisy factors make it difficult to compare the performance of predictive models, as their presence, importance and impact strongly depend on the domain (i.e. the specific organization or case for which the model was developed) [22, 41].

Consider for example the problem and domain discussed in this chapter: ‘predicting turnaround time’. In one instance of this domain, the crane operators could have the intention to offload trucks as quickly as possible, due to a financial motivation. In another, they might not have that intention at all, but could rather be focused on offloading boats, neglecting trucks. That impacts how the capacity of the equipment at the terminal is used, and therefore the amount of time it takes to service a truck. This adds noise to the data used in either case, making it difficult to compare the performance of predictive models. However, they can both be compared to the benchmark model developed for the abstract domain, as this represents a noise-free (and therefore highly simplified) version of those domains.

Arrival distribu- tion	Congested	Variables	Set	Method	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
β	Yes	C,R,S	1	Random Forest	0.488	0.490	0.491	0.492	0.493	0.502
		C,R,S	1	CART	0.452	0.454	0.456	0.456	0.457	0.461
		R,S	2	Random Forest	0.365	0.365	0.365	0.365	0.365	0.365
		R,S	2	CART	0.365	0.365	0.365	0.365	0.365	0.365
	No	C,R,S	3	Random Forest	0.452	0.466	0.467	0.467	0.475	0.477
		C,R,S	3	CART	0.447	0.453	0.454	0.456	0.459	0.468
		R,S	4	Random Forest	0.257	0.257	0.257	0.257	0.257	0.257
		R,S	4	CART	0.257	0.257	0.257	0.257	0.257	0.257
HLH	Yes	C,R,S	5	Random Forest	0.350	0.356	0.363	0.364	0.371	0.382
		C,R,S	5	CART	0.247	0.255	0.256	0.256	0.258	0.264
		R,S	6	Random Forest	0.194	0.198	0.200	0.200	0.202	0.203
		R,S	6	CART	0.193	0.193	0.193	0.193	0.193	0.193
	No	C,R,S	7	Random Forest	0.328	0.340	0.341	0.342	0.350	0.354
		C,R,S	7	CART	0.237	0.253	0.254	0.254	0.260	0.263
		R,S	8	Random Forest	0.110	0.112	0.114	0.114	0.115	0.117
		R,S	8	CART	0.108	0.108	0.108	0.108	0.108	0.108

Table 3.2
Accuracy for each of the classification tests. The best value for each dataset is shown in bold.

Arrival distribu- tion	Congested	Variables	Dataset	Method	ME	MAE	MSE	MAPE	RMSE
β	Yes	C,R,S	1	Random Forest	-21	1462	3783000	25.0	1945
		C,R,S	1	CART	-6	1151	2228000	18.0	1492
		C,R,S	1	Linear regression	-2	1156	2249000	18.0	1499
		R,S	2	Random Forest	12	2405	9006000	41.0	3000
		R,S	2	CART	-1	1570	4397000	24.0	2097
		R,S	2	Linear regression	-4	1567	4364000	24.0	2089
	No	C,R,S	3	Random Forest	0	234	85440	22.0	292
		C,R,S	3	CART	1	194	60670	17.0	246
		C,R,S	3	Linear regression	-3	194	60310	17.0	246
		R,S	4	Random Forest	0	337	169800	32.0	412
		R,S	4	CART	-5	341	173000	32.0	416
		R,S	4	Linear regression	-3	339	172000	32.0	415
High-low-high	Yes	C,R,S	5	Random Forest	-2	498	434000	14.0	659
		C,R,S	5	CART	0	353	198700	8.0	446
		C,R,S	5	Linear regression	1	352	197200	8.0	444
		R,S	6	Random Forest	-4	696	1074000	23.0	1036
		R,S	6	CART	7	591	816700	20.0	904
		R,S	6	Linear regression	6	587	811300	20.0	900
	No	C,R,S	7	Random Forest	-8	587	560100	21.0	749
		C,R,S	7	CART	-2	362	204400	11.0	452
		C,R,S	7	Linear regression	5	359	200400	11.0	448
		R,S	8	Random Forest	-2	1084	1696000	40.0	1302
		R,S	8	CART	-5	1068	1621000	39.0	1273
		R,S	8	Linear regression	-4	1066	1614000	39.0	1270

Table 3.3

Mean performance for each of the regression tests. Values are in seconds, except for MAPE, which is a percentage. The best value for each dataset is shown in bold.

The extent to which a domain is affected by factors such as behavior and politics is determined by the complexity of the domain [22,41]. The more complex a domain is, the more it will be affected by soft factors. A complex domain is one that is open to its environment; that is not fully observable; that represents a situation where people try to act together; that has parts with their own goals; that is probabilistic; or that involves politics, myths, and meanings; and that is subject to behavioral influences [59]. By extension, a simple domain is one that does not have these attributes, or where they have limited influence.

By their nature, domains in the supply chain and transport are complex domains, and the domain of a container terminal is no exception. Real world instances of this domain are strongly influenced by their environment, such as the supply of containers to the terminal and traffic conditions on the surrounding infrastructure. They represent a situation where truckers, planners, and crane operators have to work together. In a real world container terminal, truckers and crane operators have their own goals and incentives, and politics between the transport company and the terminal operators could play a significant role.

In short, a real world instance of the domain used for our benchmark is likely to be a complex domain. This means that it will be affected by noise from factors such as behavior, politics, and the environment of the domain. As our simulated container terminal represents a simple domain, our benchmark predictive models were developed with data that was not affected by any of these factors. Furthermore, it has a minimum set of variables that are likely to be found in any instance of the domain. This means that the results of the benchmark predictive models serve as a comparison for instances of the container terminal domain. Whether that comparison is good enough, depends on the complexity of the other domain: the more factors it has that are not in the simulation, the less useful the comparison would be. The benchmark is mostly useful as a comparison for domains where all variables in the simulated data play a role, but where extra noise is added.

The goal of this chapter was to demonstrate that PID-SEDA for simple domains produces predictive models with good predictive power. The reasoning being that if there are no soft factors that could create noise in the data, a soft-exclusive domain analysis would also capture all relevant factors. Looking at the performance of the predictive models, we see that the best performing model in terms of MAE and MAPE scores 352 seconds and 8 percent. That means that is only 6 minutes off on average, and that predictions are within eight percent of the actual value. Most of the other predictive models, especially those with uncongested datasets score equally well on mean absolute

error. This shows that at least for these cases, the soft-exclusive domain analysis resulted in predictive models with sufficient predictive power, as expected based on the theory from Chapter 1.

§3.5 Conclusion

In this chapter, we have demonstrated a method to create benchmarks for predictive model development. We have presented benchmark results for predicting turnaround time for a truck at a container terminal. Turnaround time is the total time it takes a truck to be serviced at a container terminal. This includes registering, unloading and/or loading a container at the terminal. The key aspect of our benchmark results is that they are based on a noise-free dataset constructed using a simulated terminal. This data source means that the predictions and predictive power are not affected by case-specific ‘noise’ such as behavioral influences and politics [3, 134].

Our simulated terminal is based on a structured review of the relevant literature. From this review, we extracted which variables are important for turnaround time, and how these are distributed. These were then combined in a simulated model of a container terminal. We used the simulated terminal to create four datasets, each with minor differences in terms of how many trucks arrive per hour, and how much time is in between the individual truck arrivals. We therefore construct a benchmark that is usable for a broad range of specific case situations with different levels of capacity (how many trucks it can handle) and congestion (how many trucks it is handling). One such case is the terminal presented in Chapter 5.

We used the four datasets to train predictive models for the purposes of both regression (predicting the exact turnaround time) and classification (predicting a range of turnaround time). The performance of the predictive models was measured with a multitude of metrics, again to make sure that the benchmark serves as a comparison for a wide range of situations.

The benchmarks developed in this chapter are especially usable as a comparison for complex domains, where behavior, myths, meaning, politics and environmental influences lead to noise, but where the variables that make up the simulated data also play a role. There, the benchmark provides clean results for comparison, that are not affected by any of these factors. This shows the potential predictive power that could be achieved for the complex domain, if all noise could be removed.

§3.5. Conclusion

4 Soft-exclusive domain analysis for a complex domain

In this chapter, we present a case study of a complex domain: a distribution center in the Netherlands. The prediction problem at this distribution center is about the arrival time of the trucks that come to pick up goods for further distribution throughout Europe, Africa and the Middle-East. The domain is strongly affected by behavior, notably that of the truckers and the truckers' planners.

The goal of this chapter is to demonstrate the effect of using a soft-exclusive domain analysis for developing a prediction instrument for truck arrival time. Based on the theory presented in Chapter 1, the hypothesis is that the soft-exclusive domain analysis will miss factors that affect the prediction goal, and it means that the domain remains partially not understood. In turn, this will have a negative affect on both the predictive power of the prediction instrument.

As noted in Chapter 1, this negative effect of using soft-exclusive domain analysis is not inherent to the method. Only when the domain the prediction instrument is developed for is complex, and the researcher is not an expert on that domain, do we theorize that soft-exclusive domain analysis will result in a lack of predictive power.

This chapter is structured as follows. First, we present an introduction into the case that is the theme of this chapter: predicting at what time a truck will arrive at a distribution center. Next comes a description of the use of the PID-SEDA method described in Section 2.3. The method starts with a structured literature review to determine the factors that influence arrival time. In the second stage of the method, those factors are used to create predictive models. In the third and final stage, one predictive model is selected based on its pre-

dictive performance. The chapter ends with a discussion of the case study and its results, and a conclusion of the extent to which the chapter has achieved the goals set out above.

§4.1 Case description

Distribution centers (DCs) are the hubs which connect transport streams in the supply chain. Trucks arrive at a DC, pick up and/or drop off goods, and continue to their next destination. Late arrivals present a problem for a DC. For example: the goods for the late trucks obstruct operations at a DC's staging areas; other trucks might be waiting for the late truck's cargo; or trucks arriving late could be in the way of trucks that arrive on time. Likewise, early arrivals might cause congestion and queues at a DC due to more trucks being on a DC's terrain than it can handle. This means both early and late arrivals prevent a DC from fully utilizing its capacity to load and unload cargo. This lack of punctuality and, more generally, synchronization between arriving and departing trucks poses a major problem for DCs [147].

To address the problem of synchronization, there needs to be a way to determine more reliably when a truck will arrive. If that is known accurately, the DC can respond adequately. Although there is a large body of work concerning DC management, and there are many studies concerning travel/journey time prediction, there are only a few papers that combine these fields.

In this case study, we investigate predicting arrival time at a DC in the Netherlands. This DC is an independent company, with several clients in the frozen goods sector. These clients send their goods to the DC, from where the client's drivers pick up the goods and transport them throughout Europe, Africa, and the Middle East.

Figure 4.1 shows an overview of the key operations at this DC. One day before travelling to the DC, truckers have to reserve a time slot at the DC. When they report at the DC, they are first sent to a queue of trucks if their slot is not due. Behind the scenes at the DC, the goods for the scheduled truck are collected from the warehouse by forklift trucks. The goods are then set up just behind the docks, from where they will be moved into the truck. As the figure shows, there is limited capacity to stage goods. This means that if a truck does not arrive in time, that staging area cannot be used, reducing the capacity of the DC. Therefore, late trucks have to be sent immediately to the dock, so the staging area can be freed up.

Figure 4.1

Schematic overview of the distribution center. Trucks wait in a queue before arriving at the docks, where they receive goods from the adjacent staging area. The staging areas are supplied from the warehouse. The last truck in the queue is late, but as its staging area is already full, it can directly go to the dock.

In this case study, we attempt to find ways to predict a truck's arrival time, as this enables the DC to respond adequately to exceptions. The DC could for example delay or speed up the cargo staging process to address a late or early arrival, respectively. Another approach to more reliable arrival times would be some method of enforcing it through company policy, but this is a) not always possible due to the nature of the relation between transport company and DC (the transport company is DC's client, or has no contractual obligation to be punctual) and b) not a substitute for prediction, as enforcement does not guarantee punctuality.

For the DC, it is important that their planning is reliable, because it dictates when the cargo has to be staged for an arriving truck and indicates the number of personnel required. Furthermore, trucks arriving late cause pile up of cargo and inefficiencies due to the order of the cargo at the docks, and truckers arriving too early cause congestion on the DC's lot. The DC tries to deal with the late arrival problem by allowing late trucks to skip the queue, so their cargo is no longer in the way of the other trucks.

The DC assumes that some truckers could abuse this policy; by arriving late, a trucker knows that he will have no waiting time at the gate. In other words, the DC is rewarding lateness which is a perverse incentive. Two solutions spring to mind: 1. penalize late arrivals, but the DC doesn't want to do this as the truckers –or more precisely the companies they work for– are in fact its customers; or 2: call the truckers and ask them to give their location. The latter is not currently feasible, as the DC does not know in advance who will be the trucker, and if they do, it is not certain they speak the same language – there are at least twelve different languages spoken among truckers at the DC.

So, the DC wants to investigate an alternative solution: predicting arrival times using historical patterns, taking into account the environmental factors (such as weather, congestion) that influence arrival time. As the DC explicitly does not want to consider the 'soft' factors such as strategies and behaviors for the reasons specified above, and therefore will not allow interviews with truckers or the like, this case study falls back to a soft-exclusive domain analysis. Therefore, we used the PID-SEDA method described in Section 2.3, to demonstrate the effect PID-SEDA has on making predictions in a complex domain.

§4.2 Stage I: Assumptions

In this first stage of prediction instrument development (PID), assumptions on what affects the prediction outcome are collected. This forms the major component of domain analysis as part of PID. As the domain analysis for this case study is soft-exclusive, it is in the form of a structured literature review.

Besides the results from the literature review, this section also presents the prediction goal and the data constraint. The latter describes what data is available for prediction, and what quality that data is. Finally, to assess the reliability of the assumptions found from the review, we describe a survey conducted among the truckers arriving at the DC. In this survey, truckers were asked to what extent each of the factors from the literature review affected their arrival time.

§4.2.1 Goal definition

The goal variable for this case study is *arrival time*, which means the time that a truck is ready to be loaded at a dock. This *actual* arrival time is predicted relative to the *scheduled* arrival time. A positive arrival time means that the truck is late, and a negative arrival time means that the truck arrives earlier than scheduled. This metric of actual time relative to scheduled arrival time is called *tardiness* in this case study.

For arrival time prediction, there are two possible goals in the terminology of Shmueli and Koppius [127]: the first is to exactly predict a numerical outcome; e.g. to predict a truck will be 37 minutes late, which is known as *regression*. The second goal type is predicting the time interval an outcome falls into, e.g. thirty to sixty minutes late, which is known as *classification*.

The DC does not require minute-level accuracy, and therefore it is sufficient to group truck arrival time into specific intervals. Therefore, the latter goal type (classification) is used in this research.

§4.2.2 Literature Review

The goal of the literature review is to determine which factors were considered to be the most important predictors for arrival time in previous research. To determine these factors, we used a structured literature review using the steps described in Webster and Watson [155].

As the papers were reviewed, the set of factors found quickly reached saturation, i.e. no new factors were mentioned in new papers. Therefore, the literature review described below does not include a forward and backward search.

Determine databases/journals. The Scopus search engine was used, as it indexes business/operations research (OR) journals, information systems (IS) journals, and relevant information technology (IT) literature. The journals that were selected for inclusion are the top journals in IS, based on the Association for Information Systems (AIS) Senior Scholar's Basket of Journals [5]. The relevant OR journals were selected based on [54]. The complete list of included journals is below.

■ Operations Research:

- Environment and Planning A
- Journal of Environmental Economics and Management
- Journal of the American Planning Association
- Journal of Transport Economics and Policy
- Journal of Operations Management
- Management Science
- Regional Studies
- Transportation Research Part A: Policy and Practice
- Transportation Research Part B: Methodological
- Transportation Research Part C: Emerging Technologies
- Transportation Research Part D: Transport and Environment
- Transportation Research Part E: Logistics and Transportation Review
- Transportation Science
- Supply Chain Management
- IEEE Transactions on Intelligent Transportation Systems

■ Information Systems:

- European Journal of Information Systems
- Information Systems Journal
- Journal of AIS
- Journal of Information Technology
- Journal of Management Information Systems
- Journal of Strategic Information Systems
- MIS Quarterly
- Information Systems Research

Query abstracts The following query on Scopus [36] was used to search the selection of journals:

(“supply chain” OR logistics OR transport OR barge OR “” OR
 freight OR ship OR truck)
 AND
 (“arrival time” OR “journey time” OR “travel time” OR delay OR tardiness OR
 lateness)
 AND
 (factor* OR variables OR influence* OR predict* OR model OR
 simulat* OR estimat* OR variab* OR reliab* OR “”)

This resulted in a list of 1801 papers, with publication dates varying from the mid-nineties to 2013.

Evaluation of abstracts After identifying the initial set of papers, their abstracts were examined to see if they are indeed relevant to the current research. Those that were not relevant were discarded. The inclusion criterion was ‘mentions arrival time or a synonym’. Those abstracts that did not mention travel time/journey time, arrival time or synonyms were all discarded. After this step, 82 papers remained.

Evaluation of papers The 82 papers were then reviewed in full. Those that do not discuss the influence some factor/a set of factors (environmental, behavioral, or other) has on travel time/journey time or a similar dependent variable were discarded. After this step, 45 papers remained.

The above literature review resulted in a list of twelve factors that are found to have an influence on travel time. Table 4.1 shows an explanation of the meaning of each of these factors.

<i>Factor</i>	<i>Description</i>
<i>Congestion/flow</i>	The amount of congestion is the inverse of the flow on a stretch of road: higher congestion leads to longer times necessary to cross that stretch. In other words, congestion leads to traffic flowing more slowly.
<i>Weather</i>	The weather conditions, such as rainfall and wind. These conditions lead to lower flow/higher congestion

Continued on next page...

<i>Factor</i>	<i>Description</i>
<i>Speed</i>	The speed at which traffic is going determines when it will arrive. Higher speeds are only possible in uncongested traffic
<i>Distance</i>	The distance a truck has to travel to its destination. Higher distances increase the probability that the truck will encounter delaying factors along the way.
<i>Type of cargo</i>	Certain types of cargo being transported lead to longer travel times.
<i>Type of truck</i>	Larger trucks take longer travel times.
<i>Time of day/week/month/year</i>	Seasonal influences as well as day-to-day peak hours influence the amount of congestion, and therefore influence travel time.
<i>Cumulative deviation</i> <i>previous</i>	The number of previous stops a vehicle has had to make and the delay it incurred there. The more stops a vehicle has to make, the higher its cumulative schedule delay might be.
<i>Accidents/incidents</i>	Incidents and accidents lead to road closures (and therefore congestion) or diversions which lead to longer travel times.
<i>Road work</i>	Road works lead to lower flow speeds (higher congestion) or diversions, therefore causing longer travel times.
<i>Traffic signal failures</i>	Signal failures lead to lower flows/speeds, and could lead to accidents.
<i>Road condition</i>	Poor road conditions lead to lower flows/speeds
<i>Driving style</i>	A conservative driving style might lead to slower speeds, whereas a more proactive style would lead to a higher speed.

Table 4.1

Explanation of the concepts in the literature review

The conclusion from the literature review and from Figure 4.2 (see also Appendix F) is that congestion, time of day/week/month/year and accidents are the most frequently mentioned factors that theoretically (i.e according to

Soft-exclusive domain analysis for a complex domain

Figure 4.2

Factors mentioned in the relevant papers from the literature review. The total number of relevant papers was 45.

the body of literature) influence travel time, and therefore are variables that should be used in the data set and the subsequent model. The majority of the papers refer to congestion as the main cause of delays. In those papers, the other factors are mentioned as factors that create more congestion, which in turn leads to longer travel times.

Interestingly, only two papers provide an empirical validation for the claim that a particular factor has a significant influence on travel time/arrival time, even though the majority of papers uses empirical data to validate the predictive power of their model as a whole. That means that most papers claim that factors like weather and time of day influence travel time (mostly as a consequence of causing increased congestion), but don't provide a validation for that claim. The claim that congestion explains travel time is validated in most papers, by testing the predictive models, which were produced with the assumption that congestion is an important predictor.

As most papers refer to some literature explaining the factors that influence travel time, those papers were checked again to see if those referred papers provide empirical evidence. One often cited paper by Van Lint *et al.* [150] does give an overview of the sources of congestion, but only congestion as an influencer of travel time is used in the paper's predictive model. The papers that use empirical data use either loop detector data or (a variant of) floating car data –measuring flow rate at a certain time and place. None of the papers use any other data sources, such as weather information.

Finally, none of the papers investigated arrival time: all papers investigated *travel time*. Arrival time is the consequence of travel time in combination with departure time, so although the travel time literature is applicable, the human factor involved in planning the time of departure can affect the arrival time (especially for truck drivers who have travelled the same route before). Therefore, there seems to be a gap in the literature regarding arrival time prediction.

§4.2.3 Survey results

To better understand whether the factors from the literature review match the trucker's perception of what causes early or late arrivals, a survey of truckers arriving at the DC was conducted. The survey was based on the factors from the literature review. The survey is built up of eight five-point scale questions on the influence of factors, and seven questions about the trucker's previous and subsequent stops.

In the survey, similar factors were grouped: congestion, accidents/incidents, road work and road condition are grouped with *traffic/congestion*, as they are all related to the road and the vehicles on that road, and congestion is the most occurring factor in the literature review; speed and driving style are grouped *driving style*. *My company's planning* was included as a factor at the request of the DC, because they assumed this would also be influencing arrival time. The survey included two lines under the “other” category to allow truckers to fill out the factors that they experienced to be influencing their arrival time. The survey was administered in 11 languages (Dutch, English, French, Spanish, German, Russian, Bulgarian, Romanian, Polish, Slovenian, and Czech), and it was filled out by truckers immediately on arrival at the DC.

The English translation of the survey is shown in Appendix G. The truckers were asked to indicate whether a factor (from the literature review) influenced their arrival time, and to indicate the extent to which each factor influenced their arrival time positively or negatively (i.e. earlier/later, respectively). The truckers were also asked to state their planned and actual arrival time, to see if there is a correlation between the factors and the trucker's tardiness. The survey was anonymous, so the trucker's stated planned arrival cannot be compared to the planned arrival communicated to the DC.

Lastly, the truckers were requested to indicate where they were driving from and where they were driving to, to gain insight into the distance and route they had to travel and, at the request of the DC, to understand the position of the DC in the chain as a whole.

The survey ran from October 1st 2013 to November 5th 2013 (35 days), during which a total of 230 responses were received. Table 4.2 shows the frequency per number of factors filled out: 8 means they were fully filled out. This shows that 124 truckers completely filled out the survey for each factor, and 31 truckers did not fill out any factor, but did fill out one or more of the other questions (about previous and future stops and so on).

Table 4.3 shows the correlations between the variables in the survey and the correlations between the independent variables and the dependent variable, tardiness (as indicated by the truckers). It also shows the significance level of each variable. This table shows that none of the independent variables are significantly correlated with the dependent variable, tardiness, but that there are significant correlations between variables¹. These findings are

¹This means that the predictors are not independent of each other, which would be problematic when using techniques such as linear or multiple regression. However, the data mining techniques used in this chapter can be used with interdependent predictors.

<i>Factors filled out</i>	<i>Frequency</i>
0	31
1	26
2	12
3	1
4	6
5	6
6	7
7	17
8	124

Table 4.2

Frequency of number of factors filled out in the survey

at odds with the result of the literature review, which suggested that congestion/traffic is the most important predictor of travel time, and therefore arrival time.

	Tardiness	Weather	Traffic	Driving style	Cargo	Planning	Distance	Truck/trailer	Time of day
Tardiness									
Weather	-0.038								
Traffic	-0.186	0.479							
Driving style	-0.092	0.390	0.064						
Cargo	-0.157	0.211	0.270	0.402					
Planning	-0.153	0.310	0.340	0.434	0.456				
Distance	-0.159	0.256	0.418	0.458	0.610	0.504			
Truck/trailer	-0.059	0.313	0.342	0.288	0.334	0.282	0.476		
Time of day	-0.078	0.324	0.399	0.299	0.546	0.380	0.546	0.602	

Table 4.3

Pearson's r correlation coefficients between the variables in the survey. Bold values indicate that those correlations are significant at the $\alpha = 0.05$ level.

§4.2.4 Data constraint

In this section, we describe the available data at the distribution center. Based on Cao *et al.*, the data constraint describes the volume, variety, veracity and velocity of available data [14]. Shmueli and Koppius [127] add *ex ante availability* as an extra aspect, which means whether data is available at the time of prediction.

The DC made planned and actual truck arrival data available covering the whole of 2012. In this period, a total of 28 189 truck arrivals were recorded. The data provided by the DC includes the planned arrival time, actual arrival time, name of transport company and destination city. Based on the literature review described in the previous subsection, besides the DC data, weather and congestion are the other major factors. This study used the weather data provided by the Royal Dutch Meteorological Institute (KNMI) [120] and traffic data provided by the Dutch bureau for waterways and highways, Rijkswaterstaat [119].

The complete dataset that can be used for making predictive models has the following set of truck, weather, and traffic variables:

- *Destination of order*: the destination country of the order. This could be the Netherlands, a neighboring country, a country in Africa, or a country in the Middle East.
- *General order information*: the order identifier and the planned arrival time. The first is a unique numerical identifier, while the second is the time and date the truck was planned to arrive.
- *Transporter information*: the transporter is the company that is responsible for the transport, the executor is the company that actually performs the transport.
- *Weather information (from KNMI)*: the mean precipitation (all kinds), mean fog, mean temperature, mean rain, maximum and minimum wind speeds
- *Traffic information (from Rijkswaterstaat)*: the maximum, minimum and mean traffic measured in number of cars per second passing a stretch of road.

§4.3 Stage II: Predictive modeling

In this second stage of prediction instrument development, the assumptions (hypotheses) from the literature review presented in the previous stage are used to create predictive models. This stage consists of developing data selection and cleaning strategies, exploratory data analysis, technique selection, the actual predictive modeling, and finally evaluation of predictive models.

§4.3.1 Data selection & cleaning strategies

Data selection and cleaning strategies determine which variables to use in a predictive model, and how these should be cleaned. Cleaning data of noise improves the potential predictive power of a predictive model developed with such a strategy.

Data cleaning strategies In developing a data selection and cleaning strategy, the variables from the literature review have to be mapped to the available data from the DC and other sources. This means that if a variable is not available in the data, a substitute has to be found. For this case study, the main missing variable in the data is truck location. Although truck location is not mentioned in the literature review, it is necessary to determine the relevant traffic and weather conditions. Therefore, we have used traffic and weather ‘zones’ as a substitute.

According to literature, traffic, weather and time of day (amongst others) influence arrival time. Of course the weather and congestion data while the truck was driving are needed, which in turn requires knowledge of where the truck was driving from, how fast it is driving and what route it is taking to the DC. None of these are available explicitly, but the frequently occurring places of departure are known from the survey. Using the Google Maps Directions Application Programming Interface (API) ^[49] the likely main routes the trucks would take were extracted, and therefore the roads the trucks would use. It was assumed that the truck would travel at the legal limit of 80 kilometers per hour. Using these two figures the position of a truck at 3, 2, and 1 hour before expected arrival can be estimated.

As the origin of a truck and its direction of travel are not known, the country was divided into three zones: zone 1 is 0-80 km from the DC; zone 2 is 80-160 km, and zone 3 is 160+ km from the DC. The reasoning is that a truck would be in zone 3 three hours before planned arrival at the DC, in zone 2 two hours before and zone 1 one hour before planned.

For each zone, the data on the road conditions for the relevant roads in that zone, as well the data from the weather stations in all three zones were aggregated. The road data is aggregated using the lowest, highest and mean score for the number of cars on any piece of relevant road in that zone, where relevance means it is part of one of the likely routes. Similarly, this study used the highest and mean score for wind speed, mean score for temperature and amount of fog and precipitation per zone. Furthermore, the study used the mean, minimum and maximum temperature, and maximum wind speed for the Netherlands as a whole.

Besides the missing variable of truck position or truck route, some variables have missing values, or values that are the result of user error. As there was no obvious replacement for missing values, the only strategy is to discard rows with missing data. After applying this strategy, 18 922 of 28 189 rows remain.

Data selection strategies Several variations of the set of factors prescribed by literature are used to test the effect of adding or changing variables on the predictive model performance. There are two main variations in the strategies described below. The first is whether the variables are made discrete (converted into categories), or are left as original (continuous). The second is whether early arrivals are left out of the dataset, so it reflects an actual dataset that would be available *ex ante*. Besides these variations, the strategies vary in which variables are used, to reflect the different combinations of variables mentioned in literature. Each of these selection strategies is combined with the data cleaning strategy described above when developing predictive models.

The selection strategies used are the following:

- S1. All variables are continuous All variables were used (weather, traffic, destination of order, general order information, transporter information) and only the dependent variable is, as mentioned before, converted to a categorical variable.
- S2. Only the traffic variables are continuous Use the same variables as for test S1, but omit (i.e. leave out) the weather variables.
- S3. Only the traffic zone 3 variables are continuous Use the same variables as for test 2, but omit the traffic information for zones 1 and 2.
- S4. Discrete traffic variables only Use the general order and transporter information, and the traffic information. The traffic variables were each split into five equal size categories from low to high.

- S5. Discrete traffic zone 3 variables only **Use the general order and transporter information, and the traffic information for zone 3 only. The traffic variables are treated the same as for test S4.**
- S6. All, weather and traffic variables are discrete **Use all the variables, where the traffic and weather variables are each split into five equal size categories, from low to high.**
- S7. Discrete weather per zone/hour and discrete traffic variables **Use the general order and transporter information plus the weather and traffic information that is zone related (not the information about the Netherlands as a whole). Weather and traffic information are split into five equal size categories, from low to high.**
- S8. Discrete weather per zone/hour and discrete traffic variables with three categories **Use the same information as the previous test S7, but with three instead of five categories.**
- S9. Discrete weather per zone/hour and discrete traffic variables with two categories **Use the same information as the previous test S8, but with two instead of three categories.**
- S10. All continuous variables, tardiness greater than -3h **Use all variables, but with only tardiness greater than -3 hours. This is to simulate prediction of truck arrival three hours early.**
- S11. Continuous traffic variables only, tardiness greater than -3h **Only the traffic variables (untransformed) are used, but only the orders with tardiness greater than -3 hours are considered.**
- S12. Continuous traffic zone 3 variables only, tardiness greater than -3h **Only the traffic variables for zone 3 (untransformed) are used, and only the orders with tardiness greater than -3 hours are considered.**
- S13. Discrete traffic variables only, tardiness greater than -3h **Use only the traffic variables (for each zone) for the orders and consider tardiness greater than -3 hours. The variables were each split into five equal size categories, from low to high.**
- S14. Discrete traffic zone 3 variables, tardiness greater than -3h **The same variables as for the previous test S13 are used, but the traffic variables for zone 1 and 2 are excluded.**
- S15. All weather and traffic variables discrete, tardiness greater than -3h **Consider variables for orders to be discrete and consider tardiness greater than -3 hours. The weather and traffic variables were each split into five equal size categories, from low to high.**

- S16. Discrete weather per zone/hour and discrete traffic variables, tardiness greater than -3h **Only the weather variables per zone (not those about the Netherlands as a whole) and the traffic variables for orders are discrete. Consider tardiness greater than -3 hours. The weather and traffic variables were each split into five equal size categories, from low to high.**
- S17. Discrete weather per zone/hour and discrete traffic variables with three categories, tardiness greater than -3h **The same variables as for the previous test S16 are used, but now the variables were split into three categories instead of five.**
- S18. Discrete weather per zone/hour and discrete traffic variables with two categories, tardiness greater than -3h **The same variables as for the previous two tests (S16 and S17), but the variables are split into two categories instead of three.**

§4.3.2 Reduction by data constraint

The data selection and cleaning strategies described above were all developed using the available dataset, to avoid developing strategies that cannot be tested/developed into a predictive model. In this section, we discuss other data constraints that could render predictive models unactionable.

Shmueli and Koppius define two constraints in choosing variables: they should be ‘available at the time of prediction’(ex ante availability) and should be good quality [127]. Other than that, they state that ‘the choice of potential predictors is often wider than in an explanatory model’.

Variables that are considered in this study are those that were identified in the literature review and survey, as described in Section 4.2, which establishes their ex ante availability and quality.

Considering the availability of these variables, there are no practical issues. Weather information is available in real-time from the Royal Dutch Meteorological Institute. Similarly, traffic information is available continuously through several outlets (but not through the source used in this chapter, Rijkswaterstaat). Some of the data selection and cleaning strategies directly take into account what data is available several hours before a truck is expected, to further reflect true availability at the time of prediction.

Quality of data is an issue: the original data is most likely reliable, but converting it to data about a ‘zone’ (see section 4.3.1) might reduce predictive power when compared to directly using the truck’s location. However, this is unavoidable, as positional data for the trucks is not available to the DC, and it is unlikely to become available in the near future.

Table 4.4

Descriptive statistics for the tardiness of trucks arriving at the DC. Negative values mean *early* arrivals, positive values mean *late* arrivals. Missing values account for the difference in total frequency and the number of rows in the original data set (which was 28189)

Tardiness	Frequency	Fraction
More than 7 hours early	1327	0.07
-7 to -4 hours	418	0.02
-4 to -3 hours	328	0.02
-3 to -2 hours	633	0.03
-2 to -1 hours	1214	0.06
-1 to 0 hours	4545	0.24
0 to 1 hours	5257	0.28
1 to 2 hours	1877	0.10
2 to 3 hours	1011	0.05
3 to 4 hours	621	0.03
4 to 7 hours	810	0.04
More than 7 hours late	881	0.05
Total	18922	1.00

§4.3.3 Exploratory data analysis

Table 4.4 shows descriptive statistics of the difference between planned and actual arrival time for all arrivals in 2012. This shows that 48 percent of trucks does not arrive within an hour of plan (late or early), and 32 percent does not arrive within two hours of plan. 18 percent is more than 4 hours early or late.

Table H.1 in Appendix H shows Pearson's r correlations between all variables² and the dependent variable: tardiness (difference between planned and actual arrival time) in the data set. This shows that there are strong correlations between many of the independent variables. Some of the traffic variables are significantly correlated with the tardiness variable, just as some of the weather variables, but the Pearson's r value for the significant relations is negligible or weak. The strongest relationship found is -0.21 between tardiness and the minimum value for traffic in zone 3, the traffic variables for zone

²max.traffic1 is maximum traffic score for zone 1, mean_t..1 is mean temperature for zone 1, mean_w..1 is mean wind for zone 1 and so on

1 have an r value of around -0.2 . The zone 2 traffic variables have an even weaker correlation. All variables other than traffic are either not significantly or negligibly correlated with tardiness.

Looking at the correlations between the independent variables, there are mostly strong and significant correlations within groups of variables (i.e. weather and traffic variables). The correlations between weather and traffic variables are mostly significant but negligible or weak in strength, with a few exceptions.

That means that in this case, linear or multiple regression cannot be applied, as there are dependencies between the predictors. Even if these dependencies could be resolved, it is to be expected that a linear or multiple regression model will result in poor predictions, as the correlation between the predictors and the dependent variable is weak.

§4.3.4 Technique selection

Several data mining techniques are used to see if it is possible to predict arrival times, even though the independent variables have a negligible correlation with the dependent variable: tardiness or difference between planned and actual arrival time. The argument for data mining is that the combination of dependent variables could be a good predictor of arrival time, and that the correlations between the independent variables make several statistical techniques inappropriate (because inter-variable independence is assumed) to use.

Among the five data mining techniques used in this study are decision trees: a variation on Breiman *et al.*'s Clustering and Regression Trees [12]; clustering techniques: k-Nearest Neighbors [142] and Support Vector Machines [32]; and ensemble techniques: Freund and Schapire's Adaboost M1 [43, 44] and Breiman's Random Forests [11]³.

This choice of techniques represents a sample of the major types of classification techniques, so any technique-specific performance issues are avoided. Also, previous research has shown that the Random Forest, SVM and Adaboost techniques are expected to perform well [134]. For each of these techniques, the R statistical software's implementation [115] was used.

Figure 4.3

Calculating the distance between the actual and predicted category. In this example, the actual category is -3 to $-2.5h$, and the predicted category is -1 to $-0.5h$. In the figure, the actual category is the first tick on the axis, and the predicted is the fifth tick. That means that there are 4 steps between the actual and predicted category. In other words, the distance is 4.

§4.3.5 Evaluation, validation & model selection

For the predictive models, performance was measured as the fraction of predictions that has a distance between actual and predicted arrival time category less than or equal to a specific threshold value which is explained below. For example: if the actual category is $-3 \dots -2.5h$ and the predicted category is $-1 \dots -0.5h$, then the ‘predicted-actual’ distance is 4 (see Figure 4.3). Since 30 categories were used, the maximum distance is 29. If there are ten predictions with the distances $[3, 4, 7, 1, 2, 4, 6, 3, 2, 1]$, then the fraction within the threshold of 3 in these predictions is $\frac{6}{10} = 0.6$.

The threshold was used as a form of reliability interval for distance predictions, and it gives a more clear indication of performance compared to just using the fraction that had the correct category. The choice of values for the threshold is based on the DC’s indication that an error of one hour in prediction would be “acceptable”.

Table 4.5 and Table 4.6 show the performance for each data selection and cleaning strategy for each prediction technique.

³An explanation of the workings of each of these techniques is considered a textbook matter, and is beyond the scope of this chapter.

<i>Test</i>		<i>Random Forest</i>	<i>CART</i>	<i>SVM</i>	<i>Adaboost.M1</i>	<i>kNN</i>
S1	All variables are continuous	0.576	0.657	0.655	0.657	0.521
S2	Only the traffic variables are continuous	0.548	0.589	0.589	0.594	0.482
S3	Only the traffic zone 3 variables are continuous	0.521	0.574	0.568	0.577	0.483
S4	Discrete traffic variables only	0.570	0.572	0.592	0.574	0.479
S5	Discrete traffic zone 3 variables only	0.567	0.570	0.569	0.568	0.477
S6	All, weather and traffic variables are discrete	0.600	0.657	0.656	0.659	0.524
S7	Discrete weather per zone/hour and discrete traffic variables	0.614	0.657	0.652	0.658	0.520
S8	Discrete weather per zone/hour and discrete traffic variables with three categories	0.636	0.657	0.655	0.659	0.521
S9	Discrete weather per zone/hour and discrete traffic variables with two categories	0.654	0.657	0.652	0.659	0.518
S10	All continuous variables, tardiness greater than -3h	0.657	0.725	0.722	0.725	0.606
S11	Continuous traffic variables only, tardiness greater than -3h	0.608	0.663	0.662	0.665	0.551
S12	Continuous traffic zone 3 variables only, tardiness greater than -3h	0.584	0.648	0.646	0.651	0.551

Continued on next page...

<i>Test</i>		<i>Random Forest</i>	<i>CART</i>	<i>SVM</i>	<i>Adaboost.M1</i>	<i>kNN</i>
S13	Discrete traffic variables only, tardiness greater than -3h	0.650	0.653	0.664	0.663	0.549
S14	Discrete traffic zone 3 variables, tardiness greater than -3h	0.637	0.640	0.643	0.642	0.553
S15	All weather and traffic variables discrete, tardiness greater than -3h	0.679	0.725	0.722	0.729	0.604
S16	Discrete weather per zone/hour and discrete traffic variables, tardiness greater than -3h	0.688	0.725	0.720	0.728	0.602
S17	Discrete weather per zone/hour and discrete traffic variables with three categories, tardiness greater than -3h	0.705	0.725	0.722	0.728	0.606
S18	Discrete weather per zone/hour and discrete traffic variables with two categories, tardiness greater than -3h	0.721	0.725	0.722	0.727	0.606
<i>Mean</i>		0.623	0.657	0.657	0.659	0.543

Table 4.5

Data mining test results. Each test result represents the mean number of predictions that had a distance ≤ 2 (or 1 hour) from the actual category. See Section 4.3.5 for an explanation of each test.

Technique	Distance														
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	
Random Forest	Best:	0.258	0.324	0.140	0.087	0.059	0.044	0.030	0.016	0.012	0.010	0.006	0.005	0.002	0.003
	Worst:	0.198	0.201	0.121	0.085	0.063	0.048	0.040	0.034	0.028	0.021	0.020	0.018	0.016	0.019
CART	Best:	0.253	0.340	0.131	0.087	0.056	0.048	0.032	0.016	0.010	0.008	0.005	0.005	0.000	0.000
	Worst:	0.203	0.255	0.113	0.078	0.051	0.043	0.031	0.025	0.018	0.015	0.014	0.012	0.010	0.011
SVM	Best:	0.263	0.325	0.135	0.085	0.058	0.047	0.027	0.017	0.012	0.008	0.007	0.005	0.003	0.000
	Worst:	0.232	0.223	0.113	0.072	0.053	0.039	0.032	0.023	0.018	0.016	0.014	0.014	0.014	0.017
Adaboost.M1	Best:	0.263	0.329	0.137	0.084	0.059	0.044	0.028	0.017	0.010	0.008	0.006	0.004	0.003	0.000
	Worst:	0.224	0.229	0.115	0.076	0.052	0.040	0.029	0.025	0.017	0.015	0.015	0.012	0.012	0.014
kNN	Best:	0.177	0.274	0.155	0.108	0.078	0.062	0.040	0.027	0.023	0.014	0.012	0.007	0.007	0.005
	Worst:	0.166	0.190	0.122	0.089	0.064	0.052	0.043	0.034	0.029	0.023	0.022	0.020	0.019	0.019
(continued)															
Technique	14	15	16	17	18	19	20	21	22	23	24	25	26		
Random Forest	Best:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
	Worst:	0.031	0.026	0.011	0.005	0.004	0.003	0.002	0.001	0.001	0.000	0.000	0.000		
CART	Best:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
	Worst:	0.029	0.067	0.006	0.003	0.002	0.001	0.001	0.000	0.000	0.000	0.000	0.000		
SVM	Best:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
	Worst:	0.045	0.044	0.010	0.004	0.003	0.001	0.001	0.000	0.001	0.000	0.000	0.000		
Adaboost.M1	Best:	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
	Worst:	0.035	0.054	0.008	0.003	0.002	0.001	0.001	0.000	0.001	0.000	0.000	0.000		
kNN	Best:	0.006	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000		
	Worst:	0.029	0.028	0.014	0.008	0.006	0.004	0.003	0.002	0.002	0.001	0.001	0.001		

Table 4.6

Fraction of predictions at specified absolute distances between predicted and actual arrival time slot in the best and worst predictive model. For each technique, the table shows the *best* and *worst* results, based on the fraction of results within that specified distance. See also: Table 4.5.

§4.4 Stage III: Model convergence

Based on Table 4.5, the best results for the techniques are around 72%. This means that 72% of the predictions are within 3 categories from the actual categories, or within 2 hours before or after arrival. Table 4.6 shows the performance of the tests in more detail, showing the number of predictions that was at distance 0, 1, 2 and so on from the actual arrival time slot in the test with the best result and the test with the worst result for each technique. The table shows that the worst predictions are more spread out than the best predictions. For example, CART's biggest distance in the best test was 11, whereas in the worst test it was 26 (not visible in the table due to rounding).

Based on the results of the data mining analysis in Table 4.5, the conclusion is that the Adaboost.M1 ensemble technique has the most accurate predictions. Not only does it have the single best prediction, its mean performance is also slightly better than the other prediction methods.

In terms of model/data selection and cleaning strategy, the best performance is achieved with the weather and traffic data split into categories. The single best performing model with the Adaboost.M1 prediction method is S15, where traffic and weather data are split into five categories. The model that has the best performance across the five prediction methods is S18, where traffic and weather data are split into just two categories.

The best performance is 72.9% of arrival predictions within 1 hour of actual arrival. Compare this to the more naive method that predicts that a new observation has average tardiness (2100 seconds). That new observation would fall into category 0.5 to 1 hour late. With this naive method, the accuracy would be 9%. At 72.9%, the data mining accuracy is higher.

§4.5 Discussion

In this study, the factors or variables that influence arrival time were investigated. Three different approaches were used for that determination: a literature review, to find the existing theory; a survey, to find the factors truckers believe to cause delays; and data analysis/mining, to see if the findings from the other approaches match the data.

The results of the literature review are clear-cut in that there is a small group of factors that influence arrival time, and the most prominent among those factors is congestion. The majority of papers found in the review point to factors other than congestion that cause delays. These stated that those other factors lead to more congestion, which leads to delays.

The trucker survey's 230 responses however, paint a different picture: none of the factors are significantly correlated ($\alpha = 0.05$) with tardiness. Truckers do point to congestion as the number one cause of large delays. Similarly, there is only a weak to negligible relation between congestion, weather and time of day in actual trucker data and their tardiness. Weather and congestion have an influence on prediction performance, but even with those variables, it is only possible to predict arrival times with 72% accuracy within two hours of actual arrival.

As mentioned in the field study section of this chapter, there is an incentive for truckers to arrive late, which may lead to strategic behaviour. For the data mining experiment, the assumption was that truckers intend to arrive at the planned time. If some of them actually have planned to arrive later – in other words, if this assumption is incorrect – the variables that influence arrival times according to literature would become worse predictors. For example, if a trucker wanted to arrive at 17:00, the traffic and weather variables leading up to 17:00 might help in predicting that truck's arrival time. If that trucker had told the DC he wanted to arrive at 14:00, those variables would not have a large influence. In other words: there is a difference in predicting travel time and in predicting arrival time.

Besides the lateness-incentive's influence, the (lack of) ability of the trucker to be timely must be considered as a factor that hinders predictive power. If a trucker simply has no idea how long a route will take, and might therefore arbitrarily leave too early or too late, this would make his arrival time less predictable. However, this might be the case for a first-time trucker, as someone who has taken the same route to the DC more than once will have a better understanding of when to leave to be on time (if he wants to be). Therefore, only a small number of the orders will be subject to this "first-time-tardiness", and it does not seem that this could explain for the overall lack of performance.

Based on the findings that congestion and weather only have a weak to negligible effect on arrival time, and considering that other organizational factors can be at play (such as negative or positive incentives), one arrives at a more elaborate model that might explain arrival time. Considering the vast body of literature emphasizing the importance of congestion on travel time and the findings from the data analysis in this chapter, it is believed that congestion could be a good predictor of travel time, and that factors like weather do have a correlation with congestion. However, time of arrival does not just depend on travel time, but also on *time of departure*. Here, organizational factors could play a role: if there is an incentive to be late, the trucker could leave later, and vice versa for an incentive to be early.

Considering the above, this case study illustrates the shortcomings of a prediction instrument developed without considering soft factors in the domain analysis. The structured literature review revealed a great variety of factors that influence arrival time, but none of the forty-eight papers mentioned anything related to incentives for arriving late. This was of course not to be expected: the incentive at the DC is the result of a combination of unfortunate circumstances specific to this one DC: the layout of the DC means full staging areas negatively affect its available capacity; and its relation with the trucking companies (they are the DC's clients) means it cannot or will not enforce punctuality.

In this case study, the lateness incentive was not captured by the domain analysis or available in the data. We have not tried to substitute it in a data selection and cleaning strategy, to demonstrate the effect of soft-exclusive domain analysis. As a consequence, the predictive power of the predictive models is lower than expected based on the literature review. This is best shown in the first column of Table 4.6: the predictive models score an accuracy of no more than 26%.

To offset this soft-exclusive analysis, let us consider the PID-CD method (described in Chapter 2) were applied. In the *Qualitative Assumptions* stage, the truckers would be interviewed to determine how important the incentive truly is. In the predictive modeling stage, effort could be made to capture the incentive in a variable, such as who the trucker is (this was not directly available from the data). The biggest contribution PID-CD could have had in this case, however, is making the domain analysis transparent: it forces the researcher to understand what factors are at play in the domain, and how that affects the prediction outcome. Regardless of whether these factors are available as data, this helps to understand why predictive performance is lower than expected.

§4.6 Conclusion

Based on a soft-exclusive domain analysis in the form of a literature review, congestion is the most important cause of delay, and is therefore influencing arrival time. Other factors such as weather conditions, accidents/incidents and the time of day are also factors that are influential according to literature, but most authors found these factors to mostly influence congestion, and in doing so, influencing arrival time. Also, the factors the literature reports on are usually not yet empirically validated. Moreover, the literature is mostly focused on travel time, and a gap exists in studying arrival time.

The literature's findings on the factors predicting arrival time are however not supported by the responses of 230 truckers in the survey that was conducted in this chapter. None of the factors were significantly correlated with tardiness, but the strongest correlation is between traffic/congestion and tardiness.

The findings from the prediction instrument development in this chapter using Shmueli and Koppius' method [127] lead to a similar conclusion. Although some of the variables are significantly correlated with tardiness, those correlations are negligible to weak. Unsurprisingly, using data mining techniques such as ensemble techniques, decision tree, clustering or support vector machines it was found that the predictive power of the model with weather and traffic was better than the model without those variables, but that the predictive power of the model is limited.

Finally, the conclusion is that although there is a large body of literature concerning travel time prediction, those findings cannot be directly applied to arrival prediction. Human factors, such as the intended time of arrival and the departure time need to be taken into account when developing an arrival prediction model. In other words, it seems that a soft-inclusive domain analysis is necessary to maximize the predictive performance, or at least to understand a lack of predictive performance.

5 Soft-inclusive domain analysis for a complex domain

This chapter presents a case study of predicting turnaround time for trucks at a container terminal. The case study presented here is therefore a real-world variant of that presented in Chapter 3, where a simulated terminal was used. The container terminal is an example of a complex domain, where political, cultural, and behavioral factors affect the prediction outcome. Therefore, this chapter uses PID-CD as a prediction instrument development method. The goal of this chapter is to demonstrate the use and effectiveness of PID-CD, and to evaluate if it is a better fit for complex domains than PID-SEDA.

The case study is performed at a transport company in the west of the Netherlands. This company operates between the Port of Rotterdam in the Netherlands and various hinterland depots. Mainly, it transports full containers from a container terminal to a depot, and then transports the empty containers back again.

For this company, it is interesting to predict what the turnaround time at a container terminal is. Turnaround time is the total time spend between entering and leaving the terminal. If this is known in advance for each terminal that needs to be visited, the schedule can be changed to minimize the total time spent at terminals. The transport company estimates that this could make the difference between doing three trips (terminal to depot and back) and four trips a day, a clearly considerable impact on its “bottom line”.

In this case study, we compare the predictive models developed through PID-CD with a predictive model developed using the PID-SEDA soft-exclusive (see Chapter 2). This is to show the difference in the hypotheses that either

approach results in, how this affects the data selection and cleaning strategies, and most importantly, what the difference is in predictive power between the two approaches.

The chapter is organized as follows. First, we present the prediction problem, and how experts were selected for domain analysis. Next, in the first stage of PID-CD, hypotheses are developed based on interviews and field study, and constraints for actionable prediction instruments are determined. In the second stage of PID-CD, the hypotheses are translated to data selection and cleaning strategies, which are subsequently used to create predictive models. In the third stage of PID-CD, one predictive model is selected as the basis for a prediction instrument. The chapter ends with a discussion of the results and what the chapter contributes to the thesis.

§5.1 Preparation

Before the core stages of PID-CD can be executed, there is a preparatory stage. In this stage, the prediction problem is identified in terms of what needs to be predicted and its complexity, and the experts are selected for Stage I's hypothesis development.

§5.1.1 Problem identification

The problem the transport company faces is predicting turnaround time at a container terminal as part of the container transport process. This means that the goal variable (i.e. what needs to be predicted) is *turnaround time*. In this case study, we investigate turnaround time at one terminal. The developed instrument can later be adapted for other terminals in the Port of Rotterdam.

Using the questions in Appendix J, we arrived at a complexity score of 3.8. This shows that the domain is complex, and therefore PID-CD is a good fit for this case.

§5.1.2 Expert selection

The transport company and the terminal companies are the organizations involved in the container transport process. Within the former, the involved actors are the planners and the truckers. Within the latter, the actors involved are the crane/equipment operators and administrative staff.

Not all actor types involved were available for PID-CD. Especially the terminal companies were hesitant to participate. One planner at the transport company noted that this was because the terminals' focus is on handling ships – something that is enforced by the ship operators, who otherwise might decide to go to another terminal instead.

In the end, we selected two planners, a manager, and a trucker from the transport company, and a member of the administrative staff of a terminal. The latter terminal is not the same as the one we are investigating, but our experts suggest that the terminals represent the same domain. With regard to the demographics of the experts, they all have more than ten years of experience in their respective positions, and are aged between thirty and sixty. For clarity, we henceforth refer to the planners as *P1* and *P2*, to the manager as *M1*, to the trucker as *T1*, and to the member of the administrative staff of a terminal as *S1*.

§5.2 Stage I

The first stage of PID-CD is about collecting hypotheses or assumptions on what influences the prediction outcome (or goal variable). The source for these hypotheses are the experts described above. Next, these are combined into one set of the most interesting or relevant hypotheses. Stage I of PID-CD ends with determining the constraints which define what prediction instruments will be actionable for this domain.

§5.2.1 Hypothesis divergence

To establish hypotheses related to turnaround time, we conducted semi-structured interviews with the planners, manager and terminal staff. Furthermore, we performed a field study with the trucker. In this study, we combined observations of the truckers actions with semi-structured interviews throughout several visits to the terminal.

This resulted in nine hypotheses, which are shown diagrammatically in Figure 5.1. The relations between hypotheses and the constructs used are listed in Table 5.1. These hypotheses show what the experts think influences turnaround time. Different experts have quite different hypotheses, often at the expense of each other. The planners and trucker for example theorize about shift changes, at which terminal operations come to a complete stop. This is not official terminal policy. On the other hand, the terminal staff member theorizes about the laziness of truckers, and how that causes a considerable increase in turnaround time. It is clear that having experts theorize sepa-

rate from each other has its benefits, as these hypotheses might not have surfaced in a combined session. The hypotheses and their sources are explained in detail below. The hypotheses and their constructs are summarized in Table 5.1.

	<i>Hypothesis</i>	<i>Constructs</i>
1	Fridays have higher turnaround time	Friday (<i>F</i>) is-a time-of-week (<i>ToW</i>), turnaround time (<i>[text]</i>)
2	Other days have lower turnaround time	Other day (<i>O</i>) is-a <i>ToW</i>
3	Every three hours is rush hour with higher turnaround time	Rush-hour (<i>RH</i>) is-a <i>ToW</i>
4	At each shift change, the terminal stops operations, resulting in higher turnaround time	Shift-change (<i>SC</i>) is-a <i>ToW</i>
5	Time of week, time of day and activities to perform (pick up, drop off, register) at the terminal influence turnaround time	<i>ToW</i> , <i>ToD</i> , Activity (<i>A</i>)
6	Mistakes in container registration results in higher turnaround time	Wrong registration (<i>WR</i>)
7	The same as 5, but the relation between constructs and turnaround time is distorted by manipulation of activity code input	<i>ToW</i> , <i>ToD</i> , <i>A</i> , Activity manipulation (<i>AM</i>)
8	The same as 7, but the relation between constructs and turnaround times is distorted by GPS inaccuracy	<i>ToW</i> , <i>ToD</i> , <i>A</i> , GPS inaccuracy (<i>GI</i>)
9	The same as 5 and 7, plus: time of day and time of week influence how many truckers are waiting together, truckers waiting together causes longer parking time, longer parking time causes higher turnaround time	<i>ToW</i> , <i>ToD</i> , Truckers together (<i>TT</i>), Parking time (<i>PT</i>).

Table 5.1

Relations between the hypotheses developed by the experts in hypothesis divergence.

(a)
Hypothesis 1

(b)
Hypothesis 2

(c)
Hypothesis 3

(d)
Hypothesis 4

(e)
Hypothesis 5

(f)
Hypothesis 6

Hypothesis 1 This hypothesis, described by planners (*P1* and *P2*) at the transport company, proposes that the time of the week influences the turnaround time. Although the planners are not always clear or consistent on how the other days of the week affect turnaround time, they think that Fridays are the days with the longest turnaround times. Their claim is that this is caused by the terminal operators ‘taking it easy’ as the end of their work week approaches.

Hypothesis 2 This hypothesis is the antithesis to the first hypothesis, and is also stated by the transport company planners *P1* and *P2*. The planners believe that there is a cyclic nature to turnaround time throughout the week, but they do not know which days or parts of days have shorter turnaround times, or why that would be.

Figure 5.1

Diagrams of the hypotheses developed during the hypothesis divergence stage. *TA* stands for turnaround time. A ‘+’ indicates that one construct *increases* the other, a ‘-’ indicates it *decreases*, and a ‘?’ indicates that the constructs are related, but the type of relation is unknown. A proposition pointing to another proposition means that it is a moderator, and its sign determines whether it increases or decreases the strength of the proposition it moderates. See Table 5.1 for an explanation of the abbreviations used.

Hypothesis 3 The third hypothesis was brought forward by the terminal staff member (*S1*). He thinks that ‘rush hour’, an event that happens multiple times a day, results in longer turnaround times. The first rush hour happens around five or six in the morning, when truckers arrive at the terminal *en masse*, so they can deliver their first container of the day at a hinterland depot at the moment it opens, usually around eight. As the truckers return with empty containers to pick up a new load, this causes the second rush hour, and so on throughout the day. As depots are spread out across the country, the staff member thinks that later rush hours are less intense than the first one in the early morning.

Hypothesis 4 The fourth hypothesis comes from the transport company’s trucker (*T1*). He thinks that shift-changes at the terminal cause delays, and therefore cause longer turnaround times. Most terminals in the port of Rotterdam operate three shifts of eight hours each. The morning shift is from eight (A.M.) to four (P.M.), the afternoon shift is from four to midnight, and the night shift is from midnight to eight A.M.. The trucker explained that in his experience, the whole terminal is shut down for thirty minutes at every shift change. During that time, no containers are loaded or unloaded. The trucker thinks this is because the terminal crane operators feel they have a right to chat with each other, and refers to that as the ‘port mentality’.

Hypothesis 5 According to the trucker (*T1*), the day of the week, the time of day, and the number of activities to perform at a container terminal combined determine the turnaround time. He believes that certain days have shorter turnaround times than others, although he does not know how big this effect is, or how constant that effect is from week to week. Furthermore, the trucker thinks that, in line with hypothesis four, the time of day makes a difference. This not only includes the aforementioned shift changes, but also other points of time during the day (such as lunch breaks) where the terminal operates more slowly or not at all.

The activities mentioned by the trucker are *registration*, where the trucker has to report at the terminal to be cleared to drop off and/or pick up a container; *drop off*, where the empty container on the truck is taken off and sent to the container stack by a large forklift called the *reach stacker*; and *pick up*, where the trucker gets a full container lowered onto his trailer from a yard crane. The trucker believes that the activities influence turnaround time in two ways: first, if there are more activities to perform, turnaround will be longer. Each of the activities is optional: sometimes there is no container to

drop off or pick up, or sometimes registration was handled remotely. Second, at some terminals, including the one discussed in this chapter, dropping off a container means that a queue of trucks that did not drop off can be skipped. This means dropping off and picking up a container reduces the total waiting time, and therefore turnaround time.

Hypothesis 6 The sixth hypothesis is from the terminal staff member (*S1*). It states that registration errors lead to truckers having to wait longer to enter the terminal, which in turn affects the turnaround time as a whole. Before picking up a container, that container has to be cleared for pick up. This means that the owner of the container (the shipper) signs off that the transport company can pick up that container. This sign-off process is registered in a port-wide information system called Portbase [114]. This same system is also used by the transport company to register which truck and trucker will be picking up the container. The container terminal notifies the transport company when a container arrives and accesses the system to see whether a trucker is indeed cleared to pick up the container.

Several things can go wrong in this process: the trucker might not fill out his credentials properly at the terminal when registering to pick up the container, the transport company might make a mistake with the truck that is to pick up the container, the shipper might not have cleared the container properly, and finally, the container might not have arrived at all.

Registration errors mean that a trucker is not cleared to pick up the container, and has to contact the transport company, which contacts the shipper. This process takes a significant amount of time. The staff member told anecdotes of truckers having to wait hours, or even days before a container was cleared. According to the staff member, some ten percent of registrations fail.

Hypothesis 7 The activities that the trucker performs are registered on the truck's computer, known as the On-Board Unit (OBU). Besides those mentioned above, these also include 'waiting', 'driving', 'pumping gas' and so on. The activities from the OBU are automatically uploaded to a fleet management system called Carrier Web [16]. Each time the truck does not move for five minutes, the OBU prompts the trucker to input what he is doing.

The trucker (*T1*) suggested that the system is sensitive to manipulation: as long as the truck keeps moving from time to time, the OBU will never prompt for an activity. Furthermore, the system only automatically records the end of an activity when the truck reaches 50 kilometers per hour. In the busy streets of the port, this could take a while. According to the trucker, the incentive for

manipulation is that truckers have to explain the time they spend on activities, and might get reprimanded if their statistics show them to be less efficient than others. This is especially the case if they have longer waiting times than other drivers.

Hypothesis 8 The primary source for data on the duration of activities at a container terminal is the truck's OBU, as described in hypothesis 7 above. As these durations might have been manipulated to better suit the trucker, the fallback option is to directly use the GPS positions recorded by the OBU. These would then be matched with the areas of the terminal where certain activities are performed. For example, the area surrounding the yard cranes would be designated the 'container pick up' area, the area where empty containers are stored is the 'drop off area', and the parking lot next to the terminal offices is the 'registration area' (see Figure 5.3).

Planners *P1* and *P2* at the transport company indicate that the GPS positions themselves are also not fully reliable. The GPS positions, like all OBU data, are sent to the CarrierWeb fleet management system. The planners believe this performs some processing on the stream of GPS positions, only retaining those that the system thinks are important. Therefore, activities determined based on those GPS positions are also potentially unreliable.

Hypothesis 9 The final hypothesis from an expert is once again concerned with the previously mentioned 'port mentality', but this time, it was stated not by a trucker, but by the terminal staff member (*S1*). Just like the trucker believed that terminal crane operators take any opportunity to 'slack off' during their shift, so does the terminal staff member think the same about truckers. In his experience, truckers that are waiting together, for example when registering to pick up a container at the terminal, tend to spend time to chat, drink coffee, and so forth. If they would immediately get back into their trucks after registering, this would reduce the total time they spend at the terminal.

From literature review: **Hypothesis 10** To compare PID-CD to soft-exclusive domain analysis, we have also included a hypothesis based on a literature review. The methods and results of the review are presented in Appendix I. This hypothesis emulates the results from stage I of PID-SEDA (see Section 2.3).

As congestion, amount of equipment, equipment capacity, gate processing rate and demand are closely related, we combine them into one construct *congestion*. Likewise, job duration, travel distance, number of containers and

reefer container are combined in the construct *route through terminal*. Arriving vessels and other modalities form the construct *other modalities*. Storage sequence and stack configuration form the construct *container location*. Operator style and operator error form the construct *operator*. Time and weather are not combined into other constructs.

Congestion, *route through terminal* and *time* are the most frequently mentioned constructs (combined 53, 7, and 6 mentions). Therefore, these form the combined hypothesis that we will test in stage II ¹.

§5.2.2 Hypothesis convergence

Hypothesis convergence is about reducing the set of hypotheses to a subset, to avoid having to test similar hypotheses more than once. Hypothesis convergence starts with two transformation steps: *specialization*, in which constructs are replaced by their subconstructs; and *standardization*, in which all synonymous constructs are replaced by one synonym. Next, the transformed hypotheses are compared, to see which hypotheses are sufficiently interesting to be tested in Stage II of PID-CD (see Section 2.2.2 for a detailed explanation of hypothesis convergence).

Hypotheses 1 through 4 are all subject to specialization of the ‘time of week’ construct, replacing it by its one sub-construct. Figure 5.2 shows these hypotheses after specialization. None of the hypotheses feature constructs that have a synonymous relation. Therefore, standardization does not lead to any changes. Hypotheses 5 through 9 are therefore unaffected by either specialization or standardization.

After these two steps, the next step is to determine the extent to which hypotheses overlap. This is done as follows: for each shared construct, the total causal influence of that construct on turnaround time is calculated (see Section 2.2.2). The hypotheses with a unique value for total causal influence between that construct and turnaround time become part of T , the set of hypotheses to be tested. Hypotheses that feature a construct that is not shared with other hypotheses also become part of T . The latter is the case for all hypotheses developed here: some do share constructs, and those shared constructs may have the same total causal influence, like time of day, activity, and time of week in hypotheses 5, 6, 8 and 9, but all of these also have a construct that is not found in any other hypothesis. In summary, for this case, hypothesis convergence does not reduce the hypotheses to be tested in Stage II.

¹These also reflect data that is tentatively available, see the *Reduction by data constraint* stage

Figure 5.2

Hypotheses one through four after specialization. In each hypothesis, the ‘time of week’ construct has been replaced by its subconstruct. The hypotheses are otherwise unchanged.

§5.2.3 Constraint definition

We asked the experts to determine constraints for actionable prediction instruments. As the transport company is the only organization that will use the prediction instrument, and the only source of data, only the planners and manager were consulted for this stage. In other words, only those that will actually use the prediction instrument determine which prediction instruments are usable and interesting.

There are four constraints that need to be determined: the *data* constraint, which states the quality and quantity of data; the *deployment* constraint, which states the infrastructural requirements on the prediction instrument; the *interestingness* constraint, which determines the minimal acceptable performance of the prediction instrument; and the *domain* constraint, which is about the integration of the prediction instrument with business rules and practices, such as the type of prediction the business needs. The domain constraint also covers any other constraints that could render a prediction instrument unactionable. The constraints for this domain are described below.

Data constraint The *data* constraint for this domain is that there are just two data sources: a table of recorded GPS positions per truck over time and a table of ‘activities’ per truck over time. The latter is from the aforementioned fleet management system. Tables 5.2 and 5.3 give an overview of the available columns in the original data for the activity data from CarrierWeb (from now

on: the CW data) and the GPS data, respectively. In total, there are 151 243 rows of CW data, and 2 594 490 rows of GPS data from all trucks of the transport company.

This data has not been filtered to only reflect the terminal discussed in this case study – an estimate is that of that total 9 000 rows of CW data and 60 000 rows of GPS data are related to this terminal. The data selection and cleaning strategies will address what amount of data from these totals is usable for predictive models. The discrepancy in table size is due to the CW data being aggregated (it is a list of activities), and the GPS data representing raw input from the OBU. Because activities are recorded less frequently outside terminals (e.g. there are longer stretches of the ‘driving’ activity), the ratio of GPS data to CW data differs for the total datasets (17 rows of GPS for 1 row of CW data) and the terminal-related datasets (7 rows of GPS for 1 row of CW data).

With regard to data quality, there are few problems on the syntactic level. There are zero rows in the CW data that have missing values, and there are 115 402 rows with missing values in the GPS (4%). Semantically, it is less clear what the quality of the available data is, but the hypotheses (especially hypothesis 6 and 8) suggest that there are likely to be problems. Again, these will be addressed by the data selection and cleaning strategies in the next stage.

Deployment constraint The transport company does not have an elaborate IT infrastructure, or processes that are integrated with information systems. Mostly, records are kept in spreadsheets, and communication with external parties is done over the phone or over e-mail. Therefore, there is truly no a system for the prediction instrument to integrate with.

There are only loose constraints in this domain regarding the time it takes to make predictions or to make predictive models, which could be crucial for some domains. The schedules for which terminals to visit are prepared one day (24 hours) ahead. That means that it is not necessary for the prediction instrument to be able to create predictive models in (near) real-time.

Considering the statement that there is no need for integration with existing infrastructure, and the predictions are not considerably time-sensitive, we will assume that there is no deployment constraint for this case.

Interestingness constraint The interestingness constraint dictates the level of performance required of the predictive models produced by the prediction instrument. That performance requirement reflects what the end users need. In

<i>Column</i>	<i>Type</i>	<i>Meaning</i>
Vehicle	Text	Vehicle identifier (license plate)
Start time	Timestamp	Start time of activity
End time	Timestamp	End time of activity
Activity	Text	The activity. There are thirty-two types of category, of which the main ones are ‘picking up container’, ‘dropping off container’, ‘disconnecting trailer’, ‘unloading’, ‘filling up tank’, ‘traffic jam’, ‘taking break’, ‘driving’, ‘waiting’, ‘resting’, and ‘loading’. Every possible activity of the truck is covered.
Start coordinates	Coordinates	Longitude and latitude where the activity started.
End coordinates	Coordinates	Longitude and latitude where the activity ended.

Table 5.2

Data available from the Carrier Web fleet management system

<i>Column</i>	<i>Type</i>	<i>Meaning</i>
OBU	Integer	Unique identifier of the on-board unit.
Fleet number	Integer	Unique identifier of the truck.
Local time	Timestamp	Time the GPS location was recorded.
Location	Text	Description of the place the GPS coordinates reference, like a town name or other geographical area.
Ignition	Boolean	Whether or not the ignition is on.
Speed	Integer	Speed of the truck at the time of recording.
Latitude/ longitude	Coordinates	GPS coordinates of the truck.

Table 5.3

Data available from the raw GPS dataset.

this case, those end users, i.e. the planners at the transport company, indicate that an error of ‘not much more than 10 minutes’ is acceptable. In other words, a predictive model that produces predictions within around 10 minutes of the actual turnaround time would be sufficient for the transport company.

When using *regression* as a prediction goal (predicting an exact outcome), this requirement is translated to the predictions having a mean absolute error (MAE) $\leq \approx 10$ minutes. Any predictive model that has higher MAE is not actionable. When using *classification* as a prediction goal (predicting a turnaround interval), the requirement does not directly translate to a metric. The requirement suggests that the intervals for classification should be ten minutes, but it is unclear how accurate the predictions should be. We assume that accuracy should not be lower than 50%, so that the majority of predictions is within 10 minutes of the actual turnaround time.

Domain constraint The domain constraint dictates the business rules and practices the prediction instrument needs to be compatible with. These include ethics and legal standards on what data is acceptable to use, but also what type of prediction to use (e.g. regression or classification).

The data that is available from the transport company does not contain the names of drivers, but only a unique identifier of the on-board unit of a truck, and the license plate of that truck. Although this means that data might be in-

directly traced back to individuals, the transport company does not consider this a problem: they already use the data from a trucker to assess the individual's performance.

Regarding other business rules or practices, the planners and manager of the transport company similarly do not see any problems. Key for them is that the prediction instrument is accurate enough, and that it can produce a numeric turnaround prediction, which can be used for scheduling purposes. This also means that either regression or classification are suitable prediction goals for this case, as though the classification results in an interval, this could still be used to determine a difference between terminals.

§5.3 Stage II

§5.3.1 Data selection & cleaning strategies

Table 5.4 shows how the hypotheses from Table 5.1 were translated to data selection and cleaning strategies. The underlying assumption for each of these strategies is that the related hypotheses are true: later testing of the predictive models will have to show whether this assumption indeed has merit. The strategies were developed using the data constraint as a guideline, so constructs were translated to available variables where possible, and variables are substituted where necessary. The details of each of the selection and cleaning strategies are described below.

<i>Hypothesis</i>	<i>Strategy</i>	
H1 H2 H3 H4	S1	Variables used are start time and turnaround time.
H5	S2	Variables used are activity (pick up or drop off), start time and turnaround time.
H6	S3	Variables used are registration problem, start time and turnaround time.
H7	S4	Variables used are activity (register, pick up or drop off), start time and turnaround time. Activity is derived from GPS data.

Continued on next page . . .

<i>Hypothesis</i>	<i>Strategy</i>	
H8	S5	The same as in S4, but terminal visits where there is more than one hour or 1500 meters between GPS reports are discarded.
H9	S6	The same as in S4, but the time for the register activity is subtracted from turnaround time.
H10*	S7	Variables used are start time, congestion, route and turnaround time. Congestion is the total number of trucks from the transport company that arrived at that start time.
Every strategy		Start time is converted to hour of the week (0 – 168). Turnaround time is the moment the terminal is left minus the time the terminal was entered.

Table 5.4

Translation of hypotheses to data selection and cleaning strategies. Hypothesis 10 is the hypothesis developed based solely on the literature review.

Strategy 1 The first four hypotheses all revolve around the construct of time, specifically a certain time of the week that causes longer or shorter turnaround times. For a predictive model, it is not important whether that time is a Friday, Tuesday, or any other day. It just means that there must be a variable which reflects the time of the week, so the data mining technique can attempt to find interesting patterns in that variable. This strategy uses the CW data, as it is considered the default dataset by the transport company.

Strategy 2 The second strategy builds on the first strategy, but reflects the importance of which activities are performed at the container terminal. The relevant activities considered here are ‘pick up’ and ‘drop off’. The third potentially interesting terminal activity ‘register’ is not recorded in the CW data. ‘Pick up’ and ‘drop off’ are recorded in the dataset as boolean (true/false) values.

Strategy 3 This strategy is different from the previous two, as its focus is on whether or not there was a registration problem. A registration problem occurs when either the shipper, transport company, or terminal has made a mistake in determining whether a container is cleared to be taken of the terminal. There is no variable available in either dataset that reflects registration problems, or that can be used as a substitute.

Strategy 4 The fourth strategy is similar to strategy two, but assumes that the activities recorded by the trucker in the CW data are unreliable due to manipulation. To substitute those activities, this strategy uses the GPS data in combination with a map of the terminal. The pick up activity duration is recorded as the time between entering and leaving the ‘pick up’ area of the terminal (closest to the yard cranes), the drop off activity is related to the area surrounding the empty container stacks. Using the GPS data, the ‘registration’ activity can also be derived, by looking at when the parking lot in front of the terminal offices was entered and left. Note that this was not possible for strategy 2, as that uses the CW data, where coordinates are not available. Figure 5.3 shows a schematic map of the layout of the container terminal with all three ‘activity’-areas.

Strategy 5 This strategy once again builds on the previous strategy. Again, GPS data and the terminal map are used to determine the activities. What is different in this strategy is that it assumes that GPS is unreliable due to how it is recorded: not all GPS records are stored, but the system performs some type of aggregation. In this strategy, a visit is discarded if it appears that too many GPS records associated with that visit were dropped by the aggregation. This means that visits where the time between subsequent GPS reports is greater than 1 hour, or where the distance between subsequent reports is greater than 1500 meters are considered invalid, and are removed from the dataset.

Strategy 6 The final strategy based on the experts’ hypotheses builds on strategy four, but once again adds an assumption. For this strategy, that assumption is that truckers being physically in the same place leads to unpredictable behavior. As the associated hypothesis states, truckers are liable to waste time chatting, drinking coffee, and the like. Therefore, in this strategy, the total time spend in the ‘registration’ area (the parking lot closest to the terminal offices) is subtracted from the turnaround time. That means that the predictive models developed with this strategy only predict the time spent for dropping off and picking up a container.

Strategy 7* This strategy is based on the hypothesis from literature review². There are some differences with regard to the variables used in this strategy as compared to the other strategies. The two variables used here that have not been discussed previously are *route* and *congestion*. The first indicates the order in which the truck performed its activities at the terminal. Mostly, this route is either ‘enter → drop off → exit’, ‘enter → pick up → exit’, or ‘enter → drop off → pick up → exit’, but other variations are possible for trucks with capacity for two containers. Congestion means the number of trucks that are at the container terminal at a certain hour of the week, such as the aforementioned ‘rush hour’. As congestion is not directly available in the dataset, we substitute it by the number of trucks of the transport company that were at the terminal at that hour of the week.

Every strategy There are two issues that are addressed for every dataset. The first is that the turnaround time itself is not recorded in either of the transport company’s datasets. Therefore, turnaround time is determined to be the time between the first and last recorded activity or GPS point within the terminal. This means that the terminal map (see Figure 5.3) is used to determine when the terminal as a whole was entered and when it was left again.

Second, the start time is converted to the hour of the week, as each hypothesis suggests that the time of the week is a predictor of turnaround time. The hour of the week therefore ranges from 0, which is midnight to one AM on Sunday, to 167, which is eleven at night to midnight on Saturday.

§5.3.2 Reduction by data & domain constraints

Table 5.5 shows the variables, source, and number of rows for each of the cleaned datasets. All required data both in terms of variables (or suitable substitutes) and volume (if the volume is too low, this could skew the predictive model performance) is available for strategies 1, 2, 4, 5, 6 and 7*. There is no data available on registration problems, so strategy 3 is not developed further.

§5.3.3 Exploratory data analysis

Figures 5.4, 5.5, 5.6 and 5.7 show the results of an exploratory data analysis (EDA) of the datasets produced by the data selection and cleaning strategies. This EDA serves to understand how well the hypotheses are reflected by the data. Figure 5.4 shows the number of trucks arriving per hour of the week.

²To differentiate this PID-SEDA strategy from the PID-CD strategies, we mark it with an asterisk throughout this chapter.

Figure 5.3

Schematic of the container terminal and the registration, container pick-up, and container drop-off areas. The time between entering and leaving each area is recorded as the time that activity took. Turnaround time is the total time between entering and leaving the terminal. Note that for simplicity, the roads in the terminal are depicted as single lane, but this is not in fact the case at the pick up and drop off areas. Therefore, it is possible for vehicles to pass each other in those places.

Dataset	Variables								Source	Rows
	Activities									
	Turnaround	Start	Pick up	Drop off	Register	Route	Congestion	Registration problems		
S1	v	v							CW	3053
S2	v	v	v	v					CW	3053
S3	v							v		—
S4	v	v	v	v	v				GPS	3053
S5	v	v	v	v	v				GPS	3022
S6	v	v	v	v	v				GPS	3053
S7*	v					v	v		CW	2014

Table 5.5

Variables used, source, and number of rows in each of the cleaned datasets. A ‘v’ indicates that a variable is selected in the strategy. The discrepancy in number of rows for some of the datasets is caused by a more strict cleaning strategy for strategy 5 and less available data for strategy 7*. There is no data available for S3, as there is no data on ‘registration problems’. CW means the source is the CarrierWeb dataset, and GPS means the source is the raw GPS dataset.

The plot is based on dataset S4, so the orders for the trucks (that is, the activities performed at the terminal) are also shown. The figure shows that there are more containers picked up at the terminal than there are dropped off, and shows that a considerable number of trucks do not appear to perform an activity at the terminal. This means there are no GPS records for that truck within the ‘pick up’ or ‘drop off’ areas of the terminal, which indicates that there are problems with either GPS accuracy or aggregation, as discussed in the data selection and cleaning strategies.

The peak of truck arrivals is on Friday, which was also mentioned in Hypothesis 1. There, the assumption was that turnaround is the longest on Friday. However, the cause of that longer turnaround was not assumed to be that the terminal is busiest on Fridays, but rather that the terminal operators work slowest on that day. Therefore, the data seems to be at odds with the hypothesis.

Figure 5.5 shows the turnaround time per order type. It shows that all order types have a similar distribution of turnaround time, although the peaks for the ‘drop off’ activity seem to be more widely spread than the others. Overall, the data does not seem to support the assumption that the activities performed at the terminal are good predictors of arrival time.

Figures 5.7 and 5.6 show the congestion at the terminal and how it affects the distribution of terminal time, respectively. As mentioned in data selection and cleaning strategy 7* (which is the basis for this data), congestion is substituted by the total number of trucks that arrived at the terminal at a certain hour of the week. That number ranges from zero to 120 trucks per hour. We have divided the congestion distribution into four equally sized parts in both figures. *Low* congestion means a value between the minimum and first quartile of the congestion distribution, *low-mid* congestion means a value between the first quartile and the median, *mid-high* means a value between the median and the third quartile, and *high* means a congestion between the third quartile and the maximum.

As Figure 5.6 shows, the differences between the turnaround distributions for each congestion amount appear to be small. Table 5.6 shows a comparison of the turnaround times per congestion type. The turnaround times for high congestion are significantly higher than those for low, low-mid, or mid-high congestion. Most interestingly, the turnaround times for low-mid congestion do not differ significantly from the turnaround times for low congestion. The comparison seems to back up the claim that congestion and turnaround time are related, although the strength of that relation remains to be seen.

Figure 5.4

Number of trucks arriving per day of the week and per order type. This plot is based on data from dataset S4. Order *D* means ‘drop off’, *P* means ‘pick up’, and *N* means ‘no activity’.

Figure 5.5

Turnaround time of trucks per order type. This plot is based on data from dataset S4. Order *D* means ‘drop off’, *P* means ‘pick up’, *N* means ‘no activity’, and *D+P* means ‘drop off and pick up’.

Figure 5.6

Turnaround time distribution per congestion level. This plot is based on data from dataset *S7**. *Low* congestion means a value between the minimum and first quartile of the congestion distribution, *low-mid* congestion means a value between the first quartile and the median, *mid-high* means a value between the median and the third quartile, and *high* means a congestion between the third quartile and the maximum.

Figure 5.7

Distribution of congestion at the terminal. This plot is based on data from dataset S7*.

	Low	Low-mid	Mid-high	High
Low	–			
Low-mid		–		
Mid-high	v	v	–	
High	v	v	v	–

Table 5.6

Comparison of turnaround time per congestion type. A ‘v’ indicates that the type in the *row* has significantly ($p < 0.10$) greater turnaround time than the type in the *column*. For example: ‘high’ congestion has significantly higher turnaround time than ‘low’ congestion.

§5.3.4 Technique & parameter selection.

For the purposes of this demonstration of PID-CD, we have used two techniques: Random Forests and Classification and Regression Trees [11, 12]. For completeness, other techniques could be used to see if these increase performance. Given the domain, we do not believe that the choice of technique has a strong influence on the relative performance of models, or that the dataset has a strong influence on the performance of the technique; in other words, as the datasets are very similar, a technique that performs better than the others on one set, will probably perform better on all of the datasets.

§5.3.5 Model training.

Models were trained using the *R* suite for statistical computing [87, 115, 143]. To ensure that the training sample does not affect the performance of the model, we used ten-fold cross validation. That means that the dataset is split into ten equally sized partitions, and that there are ten iterations of model training. In each iteration, a predictive model is trained on nine of the ten partitions, and one partition is left out as a holdout set. This holdout set is used to test the performance of the predictive model developed in that iteration. Each partition is used as a holdout set exactly once. This ensures that the performance results, which are averaged over the ten iterations of the predictive model, are valid regardless of how data was sampled.

	S1	S2	S4	S5	S6	S7*
Random Forest	0.31	0.31	0.39	0.39	0.52	0.31
CART	0.31	0.31	0.39	0.39	0.51	0.33

Table 5.7

Classification results: average accuracy for each strategy

	S1	S2	S4	S5	S6	S7*
Random Forest	1451	1425	1163	1092	880	1368
CART	1473	1416	1138	1054	848	1425

Table 5.8

Regression results: average mean absolute error for each strategy

Tables 5.8 and 5.7 show the average results for the performance metrics (respectively mean absolute error and accuracy) for each strategy. Tables 5.9 and 5.10 show a comparison of the results of the ten iterations for each predictive model, to determine by t-test which are statistically better than others. Strategy 6 performs significantly better than all other strategies (t-test, $p < 0.01$ level), both for classification and regression, and regardless of the technique used. There is no significant difference in the performance of either technique for the same data selection and cleaning strategy.

§5.3.6 Reduction by interestingness, deployment & domain constraints.

The experts determined that the mean absolute error should be ‘no more than around ten minutes’. All strategies other than strategy S6 are therefore ruled out by this constraint, as the models from those strategies score around the twenty minute mark for regression. The best performing strategy (S6) has a mean absolute error of around 14 minutes. As the constraint was formulated somewhat vaguely, we assume that the predictive models for strategy S6 for both techniques will be actionable.

By design, the domain constraint that the predictive models should be regression or classification models has also been met. It would seem however, that the regression models are more interesting in this case than the classification models. As mentioned in the constraint definition in Stage I, a classification model should score higher performance (> 0.75) before being considered reliable enough.

$\nearrow$	Random Forest							CART						
	S1	S2	S4	S5	S6	S7*		S1	S2	S4	S5	S6	S7*	
Random Forest S1														
Random Forest S2														
Random Forest S4	v	v				v		v	v				v	
Random Forest S5	v	v	v			v		v	v				v	
Random Forest S6	v	v	v	v		v		v	v	v			v	
Random Forest S7*														
CART S1														
CART S2														
CART S4	v	v	v			v		v	v				v	
CART S5	v	v	v			v		v	v				v	
CART S6	v	v	v	v		v		v	v	v			v	
CART S7*														

Table 5.9

Comparison of regression results. If a cell contains a ‘v’, that means that the predictive model in the *row* has significantly ($p < 0.05$) better performance than the predictive model in the *column*. For example: the predictive model with strategy S4 and the Random Forest technique is significantly better than the predictive model with strategy S1 and the same technique.

	<i>Random Forest</i>							<i>CART</i>						
	S1	S2	S4	S5	S6	S7*		S1	S2	S4	S5	S6	S7*	
Random Forest S1														
Random Forest S2														
Random Forest S4	v	v				v		v	v				v	
Random Forest S5	v	v	v			v		v	v				v	
Random Forest S6	v	v	v	v		v		v	v	v	v		v	
Random Forest S7*														
CART S1														
CART S2														
CART S4	v	v	v			v		v	v				v	
CART S5	v	v	v			v		v	v				v	
CART S6	v	v	v	v		v		v	v	v	v		v	
CART S7*														

Table 5.10

Classification details. If a cell contains a ‘v’, that means that the predictive model in the *row* has significantly ($p < 0.05$) better performance than the predictive model in the *column*. For example: the predictive model with strategy S4 and the Random Forest technique is significantly better than the predictive model with strategy S1 and the same technique.

There is no deployment constraint, but the models developed here would fit easily with the existing infrastructure at the transport company. Data cleaning and model training for strategy S6 can be fully automated, which means that all the transport company planners need to do to make predictions is provide the necessary data.

§5.4 Stage III

As there is only one model left, stage III consists of deciding whether that model is good enough. The manager and planners at the transport company were asked for their opinion, and on whether it is likely that a rerun of the PID-CD process (with other experts or data sources) could result in a better model. Their conclusion was that a mean absolute error of 14 minutes is acceptable, and that it is unlikely that other data would become available, or that other hypotheses would result in improvements. Mainly, they consider it unlikely that other data sources become available, so, for example, strategy S3 could be tested, or that congestion could be measured directly. The reason is that there is no information exchange between terminal and transport company, as both consider their information to be sensitive. The developed model from strategy S6 is therefore the basis for a prediction instrument. The choice of technique is arbitrary, as there is no significant difference in performance, and the predictive models from both techniques are in the actionable area defined by the other constraints.

In summary, the prediction instrument for the transport company consists of:

- The *interestingness constraint* that predictive models have a mean absolute error ‘around 10 minutes’.
- The *data constraint* that only GPS position reports and activity reports are available, and that activity reports are potentially unreliable.
- The *data selection and cleaning strategy* that variables used are activity, start time, and turnaround time. Activity and turnaround time are derived from GPS data. Registration time is subtracted from turnaround time. Start time is converted to the hour of the week.
- The *Random Forest* or *CART* model training technique, as neither significantly outperforms the other.

§5.5 Discussion

In this chapter, we have presented a case in which we developed a prediction instrument using both PID-CD, and a method that does not include soft factors, i.e. a literature review. We found that although the soft-exclusive (literature based) method (resulting in strategy 7*) does perform better than the simplest model developed using PID-CD (significant at $p < 0.01$), all other PID-CD models perform significantly better than those based on literature alone (significant at $p < 0.01$). This reinforces the views of Checkland and Forrester that a soft-exclusive method does not fit well with a complex domain [22, 41], which our case study certainly is.

We consider PID-CD a level two design theory [50]. The artifact we have presented is a method, which allows to capture relevant domain knowledge, including soft factors such as behavior, politics and the like, to make better predictive models. The results discussed above are therefore an initial validation of PID-CD as a design theory.

There are two types of knowledge the results from this chapter contribute to: Ω -knowledge, or knowledge of natural phenomena and how they are related, and Λ -knowledge, or knowledge of how artefacts should be built [50]. The results of this chapter are a contribution to the Ω -knowledge, by demonstrating the necessity of including soft factors in developing predictive models for a complex domain, and by the better understanding of this specific container terminal that the domain analysis provides. It is a contribution to the Λ knowledge, as it validates the use of PID-CD for complex domains: the predictive models it produces are shown to be significantly better than those produced without PID-CD.

To clarify what the effect of domain analysis is, compare the two best performing strategies from the expert hypothesis (S5 and S6) with the strategy from the literature review (S7*). The factors that S7* considers important for predicting turnaround have considerable overlap with those in S5 and S6: start time and activities. The difference between the two from the hypothesis and the one from literature review however, is the knowledge of data quality. In S5 and S6 the GPS data is used to determine the activities performed during a visit, rather than using the data from the on-board unit, which is assumed to be subject to intentional or unintentional manipulation. Also, S5 assumes that the GPS records of a visit are not necessarily reliable and S6 assumes that the time truckers spend at parking areas should be ignored. This is information that cannot be found in literature, but does appear to be crucial for predictive performance.

The relations between the independent variables and turnaround time are stronger in the datasets from S5 and S6 than from S7*, as the mean absolute error shows. It should be noted that the transformation in S6 (subtracting the time spent on the ‘registration’ activity from turnaround) reduces variance in the turnaround, which results in lower MAE. However, the mean absolute percentage error, which shows the percentage predictions differ from the observations, is also significantly lower for S6 than S7*.

The introduction to this chapter mentioned that the case study here is a real world variant of the terminal discussed in Chapter 3. Many of the variables that affect the real world terminal are also present in the simulated benchmark terminal, such as congestion, capacity, and the route through the terminal (the benchmark terminal is based on the same literature as strategy S7*). Furthermore, the layout of the simulated terminal is comparable to that of the real world terminal. The benchmark developed there can therefore also be used to determine how well the predictive models developed here have performed, although we cannot fully account for the effects of using a simulated environment (i.e. in spite of our best efforts, we cannot be certain that it is realistic). The average MAE for all benchmark predictive models combined is 773, which shows that the predictive performance of S6 is not just good related to the other strategies developed here, but also compared to the benchmark.

As stated in Chapter 1, predictive models are often developed with implicit domain knowledge. It is not our claim that this would lead to inferior models compared to PID-CD— but a model developed with just a literature based analysis of an abstracted domain would likely be inferior. Rather, we have set out to show in this chapter that PID-CD is a good approach to use regardless of the expertise of the predictive modeler, as it provides a method to gain expertise from the domain.

§5.6 Conclusion

In this chapter, we have presented a case study to demonstrate the use of PID-CD in practice. In this case study about turnaround time at a container terminal, we compared the predictive models from PID-CD to those resulting from a soft-exclusive domain analysis in the form of a structured literature review. Three out of five of the PID-CD predictive models performed significantly ($p < 0.01$) better than the soft-exclusive predictive model.

In the domain analysis presented in Stage I of PID-CD a total of nine hypotheses were gathered and developed with the help of experts from the transport company and container terminal. One hypothesis was developed based

on literature. These hypotheses were then translated to seven data selection and cleaning strategies as part of Stage II of PID-CD. That means that some of the hypotheses were combined into one strategy. The literature based hypothesis was kept separate, as this was used to determine what the performance difference is between PID-CD's soft-*inclusive* domain analysis and PID-SEDA's soft-*exclusive* domain analysis, i.e. a literature review.

The resulting predictive models indicate that the strategies developed based on the expert hypotheses are better at capturing the meaning of data, and lead to stronger relations between the independent variables and the goal variable turnaround time. This is shown by the performance metrics, which are significantly better for the three best expert-based strategies than for the literature based one.

6 Discussion

In this chapter, we present a discussion of the findings and results of the previous chapters. Not only are the discussion sections of previous chapters revisited, but we also discuss the synthesis of all these. Furthermore, we discuss the contribution this thesis represents for prediction instrument development.

Understanding the domain is essential for the predictive power of prediction instruments. Lack of domain understanding results in missing important predictive variables, or results in not recognizing data quality issues. If the developer does not understand the domain prior to developing the prediction instrument, then a domain analysis is required.

Existing literature on predictive model development is most often not transparent on whether the domain was analyzed, and if so, how the domain analysis was executed. When there is mention of variable selection and data cleaning, the vast majority of papers mention only a literature review as a source for what variables to use and how to clean data. This is at odds with the type of domain they discuss: in all but a few papers, behavioral, cultural, and political factors, as well as external influences and goals of sub parts of the domain are at play. Those type of factors, which previous work shows to considerably reduce predictive power [132, 134], are difficult if not impossible to capture from literature alone.

Domains where the factors mentioned above affect the prediction outcome are known as *complex* domains [59]. Part of the nature of complex domains is that it cannot be assumed that they are just instantiations of some abstract and well-known domain [22, 41]. That means that a literature review of relevant variables and data cleaning is not applicable, as these represent the abstract, simple domain. Rather, domain analysis for these domains requires that the specifics of that domain are studied intensively [22].

The factors that a domain analysis specific to the domain should uncover are those that have not been (fully) quantified or measured. These factors are therefore not available in and cannot be deduced from the data. Examples of these factors include political motives, behavior, organization culture, conflicting sub goals of parts of the domain, and many more [59]. Together, we refer to these as *soft* factors, and an analysis that captures them is therefore a *soft-inclusive* domain analysis. A domain analysis that only aims for understanding of the abstract domain, such as a literature review, is by extension a *soft-exclusive* domain analysis.

The theory on complex domains [22, 41], theory on domain modeling [14, 51], and existing predictive model development methodologies [19, p.18-19] [127, p.563, p.564] emphasize the importance of soft-inclusive domain analysis. However, none of these predictive model development methods explain how to execute domain analysis, nor do any of the domain modeling methods explain how to combine domain analysis and predictive model development. Therefore, in this thesis, we have developed a method for soft-inclusive domain analysis, that we call Prediction Instrument Development for Complex Domains (PID-CD). PID-CD marries predictive model development methodologies with domain modeling.

Besides being able to capture the soft factors that affect the prediction outcome, a key requirement for a soft-inclusive domain analysis method is that it is rigorous. That means that the analysis is thorough, that the findings of the analysis are repeatable and cover all relevant aspects of the domain, that the findings are unbiased, and that the results of the analysis are independent of the researcher. A domain analysis method should be transparent, so it is as clear as possible regarding how and what domain intelligence is collected and used in developing a prediction instrument. If the prediction instrument developer has expertise on the domain, this transparency about how that domain intelligence has been gathered is often lacking, as expertise is something that is built up through experience over time, rather than being methodically collected.

In PID-CD, domain analysis is based on collecting the expertise of the actors within the domain. This expertise is gathered in the form of hypotheses, which after having been converted to hypothesis diagrams, are directly and methodically translated to data selection and cleaning strategies, which in turn form the basis for making predictive models. Below are our arguments for the rigor of PID-CD:

- Ensuring unbiasedness: The method ensures that the collection of hypotheses is unbiased through:

Figure 6.1

Scenarios for the use of various domain analysis methods and existing domain expertise. In the figure, *DC* stands for domain complexity, measured using the assessment tool in Appendix J. *DE* stands for domain expertise, which is how well the developer understands the domain at hand.

- Selecting experts from all stakeholder groups for each organization in the domain, so that all relevant ‘voices’ from the domain are represented.

In the container terminal case from Chapter 5, this meant gathering actors/experts from the transport company as well as from a terminal

- Giving equal consideration to the hypotheses of each actor, regardless of their affiliation or assumed expertise.

Often, it might be tempting to focus on those hypotheses from the most experienced people in the domain, such as the planners in Chapter 5. As the case study shows, the predictive models developed based on their hypotheses were not among the best in terms of predictive power.

- Allowing experts to anonymously contribute hypotheses, so that hypotheses with culturally, behaviorally, or politically undesirable influences on the prediction outcome can be contributed without damage to the expert’s reputation or relations within the domain.

Such a hypothesis would be the ‘lateness incentive’ from Chapter 4, which when used, could be considered to be an abuse of the relation between the transport company and distribution center. The same principle holds for the manipulation hypotheses expressed in Chapter 5. Hypotheses like these are only expressed if the anonymity of the actor is guaranteed.

- Ensuring completeness: all hypotheses are collected, even those that are evidently not actionable, such as those for which no data is available, as every hypothesis at least contributes to a complete understanding of the domain.

This was the case for the ‘registration problem’ hypothesis in Chapter 5. Although this hypothesis is at least anecdotally supported, there was no reliable data to test it. Still, as the hypothesis which produced the best predictive model is also related to registration, at the very least that hypothesis paints part of the picture of what affects turnaround time in the domain.

- Ensuring thoroughness: in the field study, the findings from direct observation are combined with informal interviews, as suggested by [112], so unarticulated but observed hypotheses are also recorded by the researcher.

Such was the case for the hypothesis surrounding the tendency of truckers to waste time chatting: before this was explicitly articulated by a terminal staff member, it was observed as part of the field study. The same holds for the registration problem: it was evident from the field study of a trucker that these are commonplace.

- Being methodical and traceable: the constructs and propositions in the hypotheses are made explicit through diagrams, which allow for a methodical conversion to data selection and cleaning strategies. The predictive models developed based on these strategies are therefore directly traceable back to one or more hypotheses. This forms a contribution to the domain understanding *ex post*: the hypotheses that led to the best performing predictive model must be assumed to be the best available theory of what affects the prediction outcome for the domain.

In the case discussed in Chapter 5, that would be that start time and activities predict turnaround time of trucks, as long as registration is disregarded.

PID-CD represents the best of two worlds: it emphasizes the importance of those people that have a deep understanding of the domain, as a consequence of being part of it, and it manages to add rigor to how that understanding is finally translated to a strategy for data selection and data cleaning.

Nevertheless, there are still parts of PID-CD that require skill and expertise, such as what types of variables and variable distributions are best suitable for certain predictive modeling techniques, or determining how to deal with missing values. The skill or expertise required here however is not *domain* expertise, but rather skill in predictive modeling or data mining, which is independent of the domain. PID-CD utilizes that predictive modeling skill in combination with the domain understanding that the method develops in order to maximize predictive power.

Based on the above, we recognize four scenarios for prediction instrument development, that depend on the complexity of the domain and the amount of expertise the developer has. Figure 6.1 shows these four scenarios, and how they depend on both domain complexity and domain expertise. As mentioned above, and discussed in greater detail in Chapter 1 and Chapter 3, for simple domains, a structured literature review represents a rigorous and complete means of domain analysis, regardless of domain expertise. Completeness means here that all relevant intelligence or knowledge of the domain is likely to be captured. For complex domains, both existing expertise¹ and PID-CD sug-

¹That is, domain expertise that existed prior to the start of prediction instrument development.

		<i>Complete</i>				<i>Complete</i>	
		N	Y			N	Y
<i>Rigorous</i>	N	–	Exp	<i>Rigorous</i>	N	–	Exp
	Y	–	CD		Y	SEDA	CD

(a) Simple domain

(b) Complex domain

Table 6.1

Rigor and expected completeness for each domain analysis type and using domain expertise, split per domain complexity. N means ‘no’, Y means ‘yes’. *SEDA* means structured literature review or the PID-SEDA method, *Exp* means expertise, *CD* means the PID-CD method. It should be noted that domain complexity is a scale, ranging from simple to complex. As a rule of thumb, a score ≥ 3 on the complexity assessment(see Appendix J) indicates a complex domain.

gest completeness, although it is difficult to determine the level of expertise a developer has, and what level is necessary. The major difference however, lies in the fact that PID-CD promotes rigor and repeatability, and using existing domain expertise does not. This is further shown in Figure 6.1. Finally, it should be noted that although a soft-inclusive analysis promotes completeness, it can never be guaranteed that *all* important information is captured.

In that figure, rigor and the expected completeness of each domain analysis type and domain expertise are shown. Note that although this figure shows only ‘simple’ and ‘complex’ as a domain type, domain complexity is in fact a scale. We consider a domain with a score ≥ 3 on the complexity assessment score (see Appendix J) to be complex, based on the literature review presented in Figure 1.4. It is noteworthy that our claim that a structured literature review results in a complete understanding of the domain is based not only on theory such as Checkland’s Soft Systems Methodology [22] and Forrester’s System Dynamics [41], but also on the results from the case study in Chapter 3. In that chapter, we demonstrated prediction instrument development for a simple domain, where the domain analysis was solely based on a structured literature review.

The predictive power shown by those results indicates that the soft-exclusive domain analysis type is a good fit for this level of domain complexity, and indicates that no important factors were missed that affect the prediction outcome.

Had *important* factors been missed, the expectation is that a mean absolute error (MAE) below 6 minutes (or a mean absolute percentage error of 11%) would not likely have been achieved.

Similarly, the case study in Chapter 5 shows the predictive power of a prediction instrument developed using PID-CD. The metrics there are the same and therefore comparable to (although expectedly worse than) Chapter 3's benchmark predictive model for the same prediction problem: turnaround time at a terminal. The best MAE score was around 800 seconds, which is just 100 seconds more than the mean MAE of all benchmark predictive models, and better than nine out of twenty-four predictive models in Tables 3.2 and 3.3. That PID-CD would result in good predictive power was hypothesized in Chapter 1, based on the same theory mentioned above [22, 41]. In comparison, Chapter 4 used a structured literature review as domain analysis for a complex domain, which resulted in poor predictive power, caused by missing important behavioral and strategic factors affecting the prediction of a trucker's arrival time.

A claim that we do not make is that PID-CD will result in prediction instruments with better predictive power than instruments founded on the expertise of the developer. Rather, PID-CD serves as a replacement for domain expertise, for those that either do not have it, or who choose to use a prediction instrument development methodology which is thorough, transparent, repeatable, and rigorous.

§6.1 Contribution

Below, we discuss the contribution we have made with this thesis. To clarify our contribution, we use Gregor and Hevner's framework for design science research (DSR) contributions [50]. We use this framework, as the work produced in this thesis is an example of design science research: it "involves a rigorous process to design artifacts to solve observed problems, to make research contributions, to evaluate the designs, and to communicate the results to appropriate audiences." [108, p.49]. That artefact is the PID-CD methodology.

In Gregor and Hevner's DSR framework, contributions are categorized into two types of knowledge: Ω /descriptive knowledge, or "knowledge about natural phenomena and the laws and regularities among phenomena" [50, p.343], and Λ /prescriptive knowledge, or "knowledge of human-built artifacts" [50, p.343]. The types and subtypes for Ω and Λ -knowledge are shown in Figure 6.2.

The contribution to the Ω type of knowledge is represented in each of the case study chapters presented in this thesis (Chapters 3, 4, 5). There, knowledge is produced about a phenomenon (the prediction problem and its do-

Figure 6.2

The two types of knowledge created in design science research. Adapted from Gregor and Hevner (2013) [50, p.344].

main), either by a literature review (Chapters 3 and 4), or by observations from a soft-inclusive domain analysis (Chapter 5). The contribution to Ω is a better understanding of those three domains, either by showing that the structured literature review results in sound prediction results for the artificial phenomenon (the simulation) in Chapter 3, or that it is insufficient in both other case study chapters. Chapter 5 is an Ω -contribution, as a real-world container terminal is studied in detail with the goal of predicting turnaround time, which has not received attention in literature (based on the literature review). For Chapter 4, the contribution is that there are more factors to consider than its structured literature review unveiled (such as incentives to be late) when predicting arrival time.

Our contribution to the Λ type of knowledge is covered by the methodology presented in Chapter 2 and the application of that methodology in Chapter 2. The methodology presents the constructs that make up PID-CD, such as ‘hypothesis’, ‘data selection and cleaning strategy’, but also ‘prediction instrument’ itself (although the latter is first defined in Chapter 1 as part of the terminology in Section 1.3). More importantly, Chapter 2 presents a method and reasoning for developing these constructs, and in the case of hypotheses,

a means of representing these in a diagram. Chapter 5 shows how the method is applied in a real-world case. The PID-CD method as a whole forms a *nascent* design theory: it represents “operational principles [...] constructs, methods, models, design principles [and] technological rules” [50, p.342].

Coming back to Gregor and Hevner’s DSR framework [50, p.345], there are four distinct contribution types, that depend on the level of maturity of the application domain and on the level of maturity of the solution: *improvement* represents low solution maturity but high application domain maturity; *routine design* represents high solution and domain maturity; *exaptation* represents high solution maturity but low application domain maturity; and finally *invention* represents both low solution and low domain maturity.

Domain maturity represents the novelty of the problem the research addresses. As Shmueli and Koppius’ method, Gu’s method, Cao’s method, and the CRISP-DM method all discuss the importance of domain analysis without giving a method for domain analysis, which is the problem here, it is safe to say that the problem of domain analysis is well-known. Therefore, the domain maturity is *high*.

Solution maturity represents the novelty of the method used for that problem. PID-CD is a completely new combination of parts of existing methods with novel constructs and techniques. It combines

- Gu and Tang’s Intelligence Meta-Synthesis, which has previously been used for macro-economical modeling [51];
- Cao *et al.*’s method for domain intelligence and the concepts of actionability and domain constraints² [14], which are tailored for association rule mining;
- Müller’s taxonomy of causal theory, which we have broadened to include hypotheses and have adapted for hypothesis convergence [97];
- and Shmueli and Koppius’ method for predictive analytics, which has been adapted to remove redundancies with the other methods that form PID-CD [126, 127].

Furthermore, it adds two novel constructs for this context: predictive hypotheses, which serve as a way to explicate domain knowledge; and data selection and cleaning strategies, which make the process of cleaning data and selecting variables explicit as well as explicitly linking predictive models with the

²In Cao *et al.*’s work, there is also mention of using IMS for domain intelligence [14, p.71]. However, the way both methods are combined in this thesis is novel.

domain knowledge from the hypotheses. As this augmented combination of methods is novel, we consider our solution to have low maturity, even though it has been validated through a case study. The contribution type we arrive at for this thesis is therefore *improvement*.

Gregor and Hevner state that an improvement type contribution should be “judged first on its ability to clearly represent and communicate the new artifact design”^[50, p.346]. This is covered by Chapter 2, which presents the method in great detail, and Chapter 5, which shows how the method is used for a real-world case. Next, the reasons why a new, improved method is developed should be “grounded in kernel theories from the knowledge base”^[50, p.346]. In our case, those kernel theories are represented by Forrester’s System Dynamics ^[41], Checkland’s Soft Systems Methodology ^[22], and Jackson and Keys’ definition of domain complexity ^[59]. The kernel theories and the reasons they provide for developing our method were discussed in Chapter 1. Finally, “the artifact must be evaluated to provide convincing evidence of its improvement over current solutions”^[50, p.346]. This convincing evidence of improvement is shown in Chapter 5, where the predictive models developed using PID-CD significantly outperformed those developed using the broadly used existing method of (structured) literature review.

In summary, our contribution is a method that improves prediction instrument development for complex domains through providing a soft-inclusive way of domain analysis. That method is well grounded in existing theory, and there is clear evidence that it is a significant improvement of the previous state-of-the-art for domain analysis as part of prediction instrument development.

7 Conclusion

In this last of the main chapters of this thesis, we provide a conclusion of the presented work by summarily providing answers to the three main research questions and sub-questions presented in Chapter 1.

After presenting answers to the research questions, we present the limitations of the research in this thesis, such as its scope, and give directions for future work on domain analysis and the PID-CD method.

§7.1 Answers to the research questions

Below, we give a summary of the answers to each of the research questions presented throughout the chapters of this thesis. We also address the methods used to answer the questions, and the chapter or chapters that provided an answer to the question. In the answers to the main questions, we also draw conclusions for the sub-questions of those questions.

The answers to sub-questions are therefore not discussed separately, but each main question lists which sub-questions are addressed by that main question. In one case, there is an overlap: a sub-question that is answered by two of the main questions.

- How does domain analysis affect predictive power of a prediction instrument?
 - Why should domain analysis be executed for predictive modeling?

This is the first of the main research questions for this thesis. The answer to the question is split into two halves: a theoretical discussion of the reasons for domain analysis and the theoretical effect it has on predictive

power, presented in Chapter 1; and a practical validation of the findings of that discussion through three case studies presented in Chapters 3, 4, and 5.

In the theoretical background, we presented the reasons for domain analysis as part of prediction instrument development. Domain analysis forms the basis for selection variables and understanding how to clean those variables, as stated in Shmueli and Koppius' predictive analytics method [126, 127] and the Cross-Industry Standard Process for Data Mining (CRISP-DM) [19]. These variables are the predictors, and they should be selected such that they present the factors that have the strongest influence on the prediction outcome.

Besides knowing what variables to include in a predictive model, the next step is understanding the quality of data, as there are several types of noise that occur in data [134], that have the potential of reducing the predictive power of the developed prediction instrument.

The available data for a prediction problem is a representation of the domain it came from. That representation is filtered: not all aspects of the domain are represented in the data, and not all aspects of the domain are represented one-to-one. Key to domain analysis is understanding the domain and how it is filtered into data, as this allows the developer to determine a.) whether the factors that truly affect prediction outcome in the domain are represented by data, and b.) whether the filter distorts the relation between those factors and the prediction outcome.

Furthermore, the domain analysis should result in a set of constraints for the prediction instruments [14]. These constraints about data, deployment, interestingness, and domain determine whether the instrument will be *actionable*, i.e. that its predictions are usable and interesting.

In summary, an *improper* domain analysis is theorized to lead to missing important variables, neglecting to remove noise from those variables, and producing an instrument that is not actionable. What makes up a *proper* domain analysis is presented below. The case studies show that a proper domain analysis (Chapters 3 and 5) leads to good predictive power of the prediction instrument, which Chapter 5 shows to be significantly higher than that of a prediction instrument developed with a domain analysis that is thought to be *improper*. Similarly, Chapter 4 shows how such an improper analysis can easily miss important aspects of the domain, and the negative consequences it is thought to have on predictive power.

- What is the relation between domain type, domain analysis type, and predictive power?
 - What types of domain analysis exist?
 - How does each type affect predictive power?
 - How do domains differ?
 - What are the different types of domain?
 - What domain analysis type fits best for each domain type?
 - What is the relation between domain complexity and domain analysis type?

As mentioned above, an improper domain analysis is thought to negatively affect predictive power. Whether a domain analysis is *proper* is theorized in Chapter 1 to depend on both the type of domain analysis and the type of domain.

Theory describes two distinct approaches to domain analysis: *systems engineering*, which views a domain as an instantiation of an abstract and well-known domain [22,41]; and *soft systems methodology and system dynamics*, which emphasize the importance of studying the specifics of a domain, and considers each domain to be unique and incomparable [22,41]. The first type of domain analysis does not reveal the ‘soft’ factors that affect a domain, such as behavior, politics, and the complexities of people trying to act together. Therefore, we refer to that analysis type as a *soft-exclusive* domain analysis. A domain analysis of the second type is by extension a *soft-inclusive* domain analysis.

Which type should be used depends on the complexity of the domain. Jackson and Keys define a complex domain as a domain where people are trying to act together; that involves politics, myths and meanings; that is unstructured; that is not fully observable; where parts have their own goals; that is probabilistic in nature; open to its environment; and subject to behavioral influences. Consequently, a domain with the opposite attributes is a simple domain [59]. Appendix J shows a tool for assessing the complexity of a domain based on this definition.

Checkland and Forrester’s work [22,41], combined with Jackson and Keys’ definition of complexity [59], led to three propositions:

- I.) A soft-exclusive domain analysis is a good fit for a *simple* domain: though it does not capture soft factors, that it is not problematic, given that these –by definition– do not play a role in a simple domain. Therefore, a soft-exclusive domain analysis will not lead to missing variables or noise for such a domain.
- II.) A soft-exclusive domain analysis is not a good fit for a *complex domain*: it fails to capture the soft factors that affect the prediction outcome, which will likely reduce predictive power.
- III.) A soft-inclusive domain analysis is a good fit for a *complex* domain: it captures the soft factors that affect such a domain by definition. This leads to a variable selection that covers all relevant factors, and a data cleaning procedure that removes as much noise as possible.

The distinction between a good fit of domain analysis type and domain type is the same as that was presented before as the distinction between *proper/improper* domain analysis. The validation for proposition I comes in Chapter 3, which presents the use of a soft-exclusive domain analysis for a simple domain, with good predictive power as a result. Chapter 4 validates proposition II, by showing that soft-exclusive domain analysis misses variables and noise that affect a complex domain, resulting in a loss of predictive power. Chapter 5 validates both proposition II and III, by showing that a soft-exclusive domain analysis for a complex domain results in significantly lower predictive power compared to a soft-inclusive domain analysis for that same domain.

- How should domain analysis be performed for predictive model development?
 - What types of domain analysis exist?
 - What domain knowledge should be captured by domain analysis?

The answer above presents the two types of domain analysis: soft-inclusive and soft-exclusive. This answer shows *how* those types are applied and *what* types of knowledge they should uncover.

As Checkland and Forrester state, a soft-exclusive domain analysis is an attempt to explain a domain as an instance of an abstract and well-known domain [22,41]. This entails that the domain is abstracted to be more similar to that well-known domain [41]. The structured review of prediction instrument development presented in Section 1.2.3 shows that this type of analysis is most often (in 28 of the 47 papers)

implemented as a literature review. This fits with the concept of soft-exclusive domain analysis: the abstract domain should be understood, which could well be achieved by reviewing the literature on that domain.

The existing literature does not provide a transparent and rigorous method for soft-inclusive domain analysis. For those cases where we assume it has been conducted, or where it is explicitly stated that it has been conducted, very little attention is given to *how* it was implemented. In cases dealing with complex domains where soft-inclusive domain analysis is not mentioned, we assume that a domain analysis was replaced by the existing expertise of the developer.

To make the process of gathering domain intelligence for a complex domain more transparent and rigorous, and independent of pre-existing expertise of the developer, we have developed a method for soft-inclusive domain analysis for prediction instrument development: Prediction Instrument Development for Complex Domains (PID-CD). This method, based (among others) on the macro-economical modeling method IMS [51] and Shmueli and Koppius predictive analytics method [126, 127], uses hypotheses and constraints as a way of capturing domain intelligence/knowledge. The method ensures that these are complete, and ensures that the predictive power of developed models is directly traceable to the collected intelligence, so it not only leads to good predictive power, but also to better domain understanding.

The hypotheses developed in PID-CD capture six types of domain intelligence:

Data intelligence shows the volume, variety, velocity and veracity of the available data in the domain. It is closely related to *network intelligence*, which shows how that data is distributed physically and shows how data sources are linked.

Human intelligence is comprised of the prediction outcome related thoughts, experiences and beliefs of people involved in the domain. *Social intelligence* reveals what groups exist within the domain, what the dynamics within those groups are, and how they are related. *Organizational intelligence* reveals roles and structures within organizations in the domain.

Domain intelligence is the superset of these five types of intelligence, and covers any type of intelligence in the domain. Besides the five types mentioned above, this includes other factors such as laws, business rules

and business practices. All types of intelligence are synthesized into constraints, that determine which predictive models will be *actionable*: i.e. interesting and usable [14].

These constraints are a *data constraint*, which states the quantity and quality of available data, and therefore rules out predictive models that require more or better data; an *interestingness constraint*, which states the minimum performance needed from the model; a *deployment constraint*, which determines what infrastructure predictive models have to be able to interact with; and a *domain constraint*, which contains laws and business rules the predictive model should be in accordance with, but also determines what performance measure to use, and any other reason why a predictive model would not be actionable.

In PID-CD, hypotheses are developed by the actors in the domain, with the aim of capturing the six types of intelligence. These are combined with the constraints to form data selection and cleaning strategies, which are subsequently used to create predictive models.

The PID-CD method is demonstrated in Chapter 5. In that chapter, we illustrate the hypotheses developed by the method, and how these are converted to data selection and cleaning strategies. As mentioned before, we have shown that this soft-inclusive domain analysis method leads to a predictive model with significantly higher predictive power for this case when compared to a predictive model based on soft-exclusive domain analysis.

§7.2 Limitations

In this thesis, we have focused on the supply chain and transport. The reason for this is that the projects performed as part of the PhD research that led up to this thesis were all in that field. The focus of the thesis is not only shown in the case studies, which are all about transport, but also in the literature review presented in Chapter 1.

Neither the theory presented in that same chapter, nor the method presented in Chapter 2 are exclusive or even tailored to any specific field. The propositions derived as part of the theoretical background for this thesis are therefore not limited in scope to any field. The same holds for the lack of existing domain analysis techniques.

However, the findings from the literature review of the use of domain analysis *are* specific to transport and the supply chain. Therefore, we cannot claim that other fields represent the same distribution of domain complexity, or that

they have a similar fraction of the use of soft-exclusive domain analysis to soft-inclusive domain analysis. A well-known type of research methodology in the supply chain and transport field is *operations research*. This research methodology is known for creating mathematical abstractions of the domain [41], so that optimization techniques can be applied to it: an example of a soft-exclusive type of domain analysis. For other fields, this research methodology might not be as prominent.

Domains in this field are also inherently complex: by being part of a supply chain, they are open to their environment, and because people are involved in many aspects of the supply chain and transport, they are also likely to be subject to behavioral, political, and cultural influences.

The propositions from Chapter 1 have been validated with case studies from the transport field. Potentially, for other fields, the impact of missing variables or noise could have a smaller effect than that seen here. Therefore, the claims we make on the relation of domain analysis, domain complexity, and predictive power have to be limited to this field.

A further limitation is the validation of the PID-CD method. All case studies have been performed by the author, as has the development of the PID-CD method. It is therefore possible that the findings presented here are subject to the personal judgment of the researcher¹, although we do not doubt the (theoretical) rigor and repeatability of the method (see Chapter 6). External validation is needed to determine for certain that PID-CD results in good predictive power for complex domains regardless of the predictive model developer. In other words, external validation will show whether PID-CD is also a rigorous and repeatable method *in practice*.

§7.3 Future work

In this section, we present directions for further research based on the PID-CD method and the relation between domain and predictive power. Where possible, we will provide an agenda below. As mentioned above, future work includes further validation of the PID-CD method in the form of applications in fields other than the supply chain and in the external validation of the method. In this section, we also address ideas on where the PID-CD method could lead to.

¹I.e. in terms of predictive modeling expertise, domain expertise, or any other way

		<i>Validation</i>	
		Internal	External
<i>Field</i>	Supply chain	Ch.5	–
	Other	–	<i>Planned</i>

Table 7.1

Validation of the PID-CD method.

§7.3.1 Validation

Table 7.1 shows three areas for further validation of the claims presented in this thesis. As mentioned before, the strength of the effect of an improper fit of domain analysis type to domain complexity needs to be investigated for fields other than the supply chain. Furthermore, the rigor of PID-CD needs to be further established through external validation, which is in progress at the time of writing this thesis.

This type of validation fits with what Venable *et al.* call “*evaluating the formalized knowledge about a designed artifact’s utility for achieving its purpose*”^[152, p.325]. In other words, it is about validating the design theory that makes up PID-CD. Venable *et al.* define two main characteristics that are used to determine an evaluation strategy: *evaluation type*, which could either be naturalistic or artificial, and reflects whether the artefact is evaluated in a lab-type setting, or in an empirical (real world) setting; and *evaluation time*, which is either *ex ante*, where a design or method is evaluated, or *ex post*, where an artefact created by the method is evaluated^[152].

PID-CD is a *socio-technical* artefact, as it requires human interaction to be used. Therefore, an ex-post evaluation of PID-CD requires that naturalistic evaluation be used: only when a real world prediction problem is addressed, can the method’s qualities be shown. Therefore, an external evaluation of PID-CD would be best addressed by case study or action research^[152].

§7.3.2 Directions

Besides the validation, there are two main areas where we believe the PID-CD method could be further augmented. These are described below.

Predictive modeling expertise One of the goals of PID-CD is to make the development of powerful predictive models independent of domain expertise. This is largely achieved through stages I and II of PID-CD (See Chapters 2 and 6). As

mentioned in the discussion, PID-CD however still requires skill and expertise in predictive modeling, such as determining which techniques to use, and how to perform extra data cleaning based on an exploratory data analysis. A possible research direction would be to try to remove the need for this expertise as much as possible, by including it in the PID-CD method.

Automation A more far-stretching goal related to the previous goal is to automate the whole process of domain analysis and predictive model development as far as possible. That would mean that creating powerful prediction instruments would be significantly simplified. That would require first removing the need for predictive modeling expertise (see above), and second, developing tools for hypothesis divergence and convergence.

Appendices

A	Summary	168
B	Samenvatting	172
C	Publications	176
D	Bibliography	178
E	Details of the domain analysis review	194
F	Details of the arrival time review	196
G	Distribution center survey	200
H	Distribution center inter variable correlations	202
I	Details of the turnaround time review	206
J	Domain complexity assessment	210
	J.1 Calculating domain complexity	211
K	List of journals	212
L	Concise method handbook	218
M	Design & layout	222
	M.1 About the cover	223

A Summary

Developing prediction instruments requires understanding the specific context they are developed for. This context, which is known as the domain, contains every factor that influences what is to be predicted, or prediction outcome. There are two key purposes for domain understanding: variable selection, which is about determining what factors are the best predictors of the outcome; and data cleaning, which is about determining what the quality of the available data is.

The more complex the domain is, the more factors affect the prediction outcome. A domain is complex if it is subject to behavioral influences, if it involves politics, myths and meanings, if its parts have goals of their own, if it is probabilistic, if it is open to its environment, and if it is not fully observable.

For a simple domain, that has the opposite attributes, a literature review would suffice as a domain analysis. The argument is that papers on comparable simple domains are likely to discuss all important variables that affect the prediction outcome, and lack of data quality (or *noise*) is expected to be less of a concern for such domains.

For a complex domain, there is some consensus that it is important that the analysis is specific to that domain. The complexity of the domain implies that there are likely to be factors at play in the domain that are unique to that domain. A literature review of similar domains is therefore unlikely to uncover *all* important variables.

What method should be used for analyzing complex domains is unclear from theory, but it is clear that it should capture the ‘soft’ aspects that are specific to the domain, such as human behavior. The type of analysis needed

is what we call a *soft-inclusive* domain analysis, and an analysis that does not capture the domain-specific factors (such as a literature review) is called a *soft-exclusive* domain analysis.

To bridge this gap, we have developed a method for prediction instrument development that uses soft-inclusive domain analysis. This method, *Prediction Instrument Development for Complex Domains* (PID-CD) starts with asking those directly involved with the domain to brainstorm on what affects what is to be predicted. Combined with observations from a field study, this leads to a set of testable hypotheses. These are combined with a set of constraints, which determine the conditions under which a predictive model is *actionable*, i.e. usable and interesting.

The hypotheses are converted to data selection and cleaning strategies, that determine which variables to use in a predictive model, and how *noise* should be removed from these variables. The constraints determine which strategies are converted to predictive models, and which predictive models have sufficient predictive performance. The domain experts and decision makers finally determine which predictive model will be used as the basis for a prediction instrument.

We have validated the theory and method above through three case studies. In the first case study, we have used a literature review as domain analysis of a simple domain. This simple domain, a simulation of a container terminal, was engineered in such a way as to be realistic, but to be free of behavioral influences and data noise. The predictive performance of the resulting models shows that the literature review captured the important factors.

In the second case study, we again used a soft-exclusive domain analysis, but this time for a complex domain: predicting arrival time at a distribution center. The literature review used in this case study failed to bring up some key factors, such as an incentive for truckers to be late. The same goes for issues with data quality. In summary, this case study showed that using a soft-exclusive domain analysis for a complex domain is likely to result in poor domain understanding, and as a consequence, poor predictive performance.

The third and final case study uses the PID-CD method for predicting the time a truck spends at a container terminal. The results from PID-CD were compared with those from a literature-based domain analysis. This shows that the PID-CD method led to models with significantly higher predictive power than those developed with soft-exclusive domain analysis.

The main contribution of this thesis is a rigorous and transparent method for domain analysis as part of prediction instrument development. We have demonstrated that this method of soft-inclusive domain analysis leads to better predictive power than would be achieved with soft-exclusive domain analy-

sis, through having a more complete view of what factors in the domain affect the prediction outcome. Furthermore, this method makes explicit what hypotheses exist in the domain, and how these lead to predictive models. This allows one to directly relate predictive power with the hypotheses, contributing to better domain understanding.

B Samenvatting

De ontwikkeling van voorspellingsinstrumenten vereist begrip van de specifieke context waarvoor zij worden ontwikkeld. Deze context, die ook wel domein wordt genoemd, bevat iedere factor die van invloed is op wat voorspeld moet worden, ofwel de uitkomst. De twee voornaamste doelen van domeinbegrip zijn: variabelenselectie, wat draait om het vaststellen welke factoren de beste voorspellers zijn van de uitkomst; en data-opschoning, wat draait om de kwaliteit van de beschikbare data.

Hoe complexer een domein is, hoe meer factoren van invloed zijn op de voorspellingsuitkomst. Een domein is complex als het onderworpen is aan gedraginvloeden, als politiek, mythes, en betekenissen van invloed zijn, als de onderdelen hun eigen doelen hebben, als het probabilistisch is, als het open staat voor zijn omgeving, en als het niet volledig observeerbaar is.

Een literatuuronderzoek voldoet als domein-analyse voor een eenvoudige domein, dat de tegenovergestelde attributen heeft. Het argument daarvoor is dat artikelen over vergelijkbare eenvoudige domeinen waarschijnlijk alle belangrijke variabelen die van invloed zijn op de voorspellingsuitkomst beschrijven, en dat data-kwaliteit (of *ruis*) naar verwachting een minder grote zorg is voor zulke domeinen.

Voor complexe domeinen bestaat enige consensus dat het belangrijk is dat de analyse specifiek is voor dat domein. De complexiteit van het domein betekent dat het waarschijnlijk is dat er factoren van invloed zijn op het domein die uniek zijn voor dat domein. Een literatuuronderzoek van vergelijkbare domeinen zal daarmee waarschijnlijk niet *alle* belangrijke variabelen bevatten.

Welke methode gebruikt moet worden voor de analyse van complexe domeinen is onduidelijk uit de theorie, maar het is duidelijk dat het de ‘zachte’ factoren die specifiek zijn voor het domein vast moet leggen, zoals menselijk

gedrag. Het soort analyse dat nodig is, is wat we een zacht-inclusieve analyse noemen, en een analyse die niet de domein-specifieke factoren vastlegt (zoals een literatuuronderzoek) wordt een zacht-exclusieve analyse genoemd.

Om dit gat te overbruggen, hebben we een methode ontwikkeld voor de ontwikkeling van voorspellingsinstrumenten die gebruik maakt van zacht-inclusieve domeinanalyse. Deze methode, Voorspellingsinstrument-ontwikkeling voor Complexe Domeinen (VIO-CD) begint met het vragen aan de direct betrokkenen in het domein om te brainstormen over wat van invloed is op wat voorspeld moet worden. Samengevoegd met observaties uit veldonderzoek leidt dit een verzameling van testbare hypothesen. Deze worden gecombineerd met een verzameling beperkingen, die de omstandigheden vastleggen waaronder een voorspellend model actiegericht is, d.w.z. bruikbaar and interessant.

De hypothesen worden omgevormd tot data-selectie- en opschonings-strategieën, die bepalen welke variabelen gebruikt worden in een voorspellend model, en hoe *ruis* verwijderd moet worden uit deze variabelen. De beperkingen leggen vast welke strategieën worden omgezet in voorspellende modellen, en welke voorspellende modellen voldoende voorspellend vermogen hebben. De experts and beslissers uit het domein bepalen uiteindelijk welk voorspellend model gebruikt wordt als basis voor een voorspellingsinstrument.

We hebben de bovenstaande theorie en methode gevalideerd door middel van drie onderzoeken. In het eerste onderzoek hebben we een literatuuronderzoek gebruikt als domeinanalyse voor een eenvoudige domein. Dit domein, een simulatie van een containerterminal, is op zo'n manier ontworpen dat deze realistisch is, maar toch vrij van gedragsinvloeden of data-ruis. Het voorspellend vermogen van de resulterende modellen laat zien dat het literatuuronderzoek de belangrijke factoren naar voren heeft gebracht.

In het tweede onderzoek hebben we opnieuw een zacht-exclusieve domeinanalyse gebruikt, maar ditmaal voor een complexe domein: het voorspellen van aankomsttijd bij een distributiecentrum. Het literatuuronderzoek in deze case study wist niet alle relevante factoren naar voren te brengen, zoals een prikkel voor de chauffeurs om te laat te komen. Datzelfde geldt voor problemen met data-kwaliteit. Samengevat laat dit onderzoek zien dat het gebruik van een zacht-exclusieve domeinanalyse voor een complex domein waarschijnlijk leidt tot matig begrip van het domein, met matig voorspellend vermogen tot gevolg.

Het derde en laatste onderzoek gebruikt de VIO-CD methode voor het voorspellen van de tijd die een vrachtwagen doorbrengt op een containerterminal. De resultaten van VIO-CD zijn vergeleken met die van een literatuur-gebaseerde

domeinanalyse. Hieruit blijkt dat VIO-CD leidt tot modellen met significant groter voorspellend vermogen dan die modellen die zijn ontwikkeld met een zacht-exclusieve domeinanalyse.

De voornaamste bijdrage van dit proefschrift is een strikte and transparante methode voor domeinanalyse als onderdeel van de ontwikkeling van voorspellingsinstrumenten. We hebben laten zien dat deze methode van zacht-inclusieve domeinanalyse leidt tot groter voorspellend vermogen dan bereikt zou worden met een zacht-exclusieve domeinanalyse, doordat het een completer beeld geeft van de factoren in het domein die van invloed zijn op de uitkomst van de voorspelling. Verder maakt deze methode expliciet welke hypothesen bestaan in het domein, en hoe deze leiden tot voorspellende modellen. Zodoende kan voorspellend vermogen direct worden gerelateerd met een hypothese, wat bijdraagt aan beter begrip van het domein.

C Publications

- [130] Sjoerd van der Spoel. “Methods for making predictions in cross-chain contexts”. In: *Cross-Chain Collaboration in the Fast Moving Consumer Goods Supply Chain*. Ed. by A.G. de Kok, J. van Dalen, and J. van Hillegersberg. 2015.
- [131] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. “A Benchmark for Predicting Turnaround Time for Trucks at a Container Terminal”. In: *Proceedings of the Big Data Interoperability for Enterprises Workshop*. Ed. by Maria Iacob and Marten van Sinderen. 2016.
- [132] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. “Predictive Analytics for Truck Arrival Time Estimation: A Field Study at a European Distribution Center”. In: *International Journal of Production Research* (forthcoming) (2015).
- [133] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. “The role of domain analysis in prediction instrument development”. In: *Proceedings of the Big Data Interoperability for Enterprises Workshop*. Ed. by Maria Iacob and Marten van Sinderen. 2016.
- [134] Sjoerd van der Spoel, Maurice van Keulen, and Chintan Amrit. “Process prediction in noisy data sets: a case study in a Dutch hospital”. In: *Data-Driven Process Discovery and Analysis*. Springer, 2013, pp. 60–83.

D Bibliography

- [1] Angelo Alessandri et al. “Modeling and feedback control for resource allocation and performance analysis in container terminals”. In: *Intelligent Transportation Systems, IEEE Transactions on* 9.4 (2008), pp. 601–614.
- [2] Behnam Amini, Jalil Shahi, and Siamak A Ardekani. “An observational study of the network-level traffic variables”. In: *Transportation Research Part A: Policy and Practice* 32.4 (1998), pp. 271–278.
- [3] Chintan Amrit, Maya Daneva, and Daniela Damian. “Human factors in software development: On its underlying theories and the value of learning from related disciplines. A guest editorial introduction to the special issue”. In: *Information and Software Technology* 56.12 (2014), pp. 1537–1542.
- [4] Constantinos Antoniou, Haris N Koutsopoulos, and George Yannis. “Dynamic data-driven local traffic state estimation and prediction”. In: *Transportation Research Part C: Emerging Technologies* 34 (2013), pp. 89–107.
- [5] Association for Information Systems. *Senior Scholars’ Basket of Journals*. 2014. URL: <http://aisnet.org/?SeniorScholarBasket>.
- [6] Prabir Bagchi, Miguel A Lejeune, and A Alam. “How supply competency affects FDI decisions: Some insights”. In: *International Journal of Production Economics* 147 (2014), pp. 239–251.
- [7] John Bates et al. “The valuation of reliability for personal travel”. In: *Transportation Research Part E: Logistics and Transportation Review* 37.2 (2001), pp. 191–229.

- [8] Michael GH Bell. “A game theory approach to measuring the performance reliability of transport networks”. In: *Transportation Research Part B: Methodological* 34.6 (2000), pp. 533–545.
- [9] Elliot Bendoly, Anandhi Bharadwaj, and Sundar Bharadwaj. “Complementary Drivers of New Product Development Performance: Cross-Functional Coordination, Information System Capability, and Intelligence Quality”. In: *Production and Operations Management* 21.4 (2012), pp. 653–667.
- [10] Kenneth K Boyer and Markham T Frohlich. “Analysis of effects of operational execution on repeat purchasing for heterogeneous customer segments”. In: *Production and Operations Management* 15.2 (2006), pp. 229–242.
- [11] Leo Breiman. “Random forests”. In: *Machine learning* 45.1 (2001), pp. 5–32.
- [12] Leo Breiman et al. *Classification and regression trees*. CRC press, 1984.
- [13] Wouter Buckinx and Dirk Van den Poel. “Customer base analysis: partial defection of behaviourally loyal clients in a non-contractual FMCG retail setting”. In: *European Journal of Operational Research* 164.1 (2005), pp. 252–268.
- [14] Longbing Cao et al. *Domain Driven Data Mining*. 1st. Springer Publishing Company, Incorporated, 2010. ISBN: 1441957367, 9781441957368.
- [15] Mei Cao and Qingyu Zhang. “Supply chain collaborative advantage: A firm’s perspective”. In: *International Journal of Production Economics* 128.1 (2010), pp. 358–367.
- [16] CarrierWeb. Mar. 16, 2016. URL: <http://www.carrierweb.com>.
- [17] Hilmi Berk Celikoglu. “Flow-based freeway travel-time estimation: A comparative evaluation within dynamic path loading”. In: *Intelligent Transportation Systems, IEEE Transactions on* 14.2 (2013), pp. 772–781.
- [18] Bongsug Chae, David Olson, and Chwen Sheu. “The impact of supply chain analytics on operational performance: a resource-based view”. In: *International Journal of Production Research* ahead-of-print (2013), pp. 1–16.
- [19] P. Chapman et al. *CRISP-DM 1.0*. 2000.
- [20] Kathy Charmaz. *Constructing grounded theory*. Sage, 2006.

- [21] Kyriakos C Chatzidimitriou and Andreas L Symeonidis. “Data-mining-enhanced agents in dynamic supply-chain-management environments”. In: *Intelligent Systems, IEEE* 24.3 (2009), pp. 54–63.
- [22] Peter B Checkland. “Soft systems methodology”. In: *Human systems management* 8.4 (1989), pp. 273–289.
- [23] Anthony Chen and Zhong Zhou. “The α -reliable mean-excess traffic equilibrium model with stochastic travel times”. In: *Transportation Research Part B: Methodological* 44.4 (2010), pp. 493–513.
- [24] Argon Chen and Jakey Blue. “Performance analysis of demand planning approaches for aggregating, forecasting and disaggregating interrelated demands”. In: *International Journal of Production Economics* 128.2 (2010), pp. 586–602.
- [25] Gang Chen, Kannan Govindan, and Zhongzhen Yang. “Managing truck arrivals with time windows to alleviate gate congestion at container terminals”. In: *International Journal of Production Economics* 141.1 (2013), pp. 179–188.
- [26] Gang Chen et al. “Terminal appointment system design by non-stationary M (t)/E k/c (t) queueing model and genetic algorithm”. In: *International Journal of Production Economics* 146.2 (2013), pp. 694–703.
- [27] Xiaoming Chen, Xuesong Zhou, and George F List. “Using time-varying tolls to optimize truck arrivals at ports”. In: *Transportation Research Part E: Logistics and Transportation Review* 47.6 (2011), pp. 965–982.
- [28] Jao-Hong Cheng, Huei-Ping Chen, and Yi-Min Lin. “A hybrid forecast marketing timing model based on probabilistic neural network, rough set and C4. 5”. In: *Expert systems with Applications* 37.3 (2010), pp. 1814–1820.
- [29] Elia V Chepaitis. “Soft barriers to ICT application in development: Trust and information quality in Russia”. In: *Journal of International Development* 14.1 (2002), pp. 51–60.
- [30] Hoi-Ming Chi et al. “Modeling and optimizing a vendor managed replenishment system using machine learning and genetic algorithms”. In: *European journal of operational research* 180.1 (2007), pp. 174–193.
- [31] Stephen Clark and David Watling. “Modelling network travel time reliability under stochastic demand”. In: *Transportation Research Part B: Methodological* 39.2 (2005), pp. 119–140.

- [32] C. Cortes and V. Vapnik. "Support-Vector Networks". In: *Machine Learning* 20 (1995), pp. 273–297.
- [33] Thomas H Davenport, Robert G Eccles, and Laurence Prusak. "Information Politics". In: *Sloan Management Review* (1992).
- [34] Philip Doganis, Eleni Aggelogiannaki, and Haralambos Sarimveis. "A combined model predictive control and time series forecasting framework for production-inventory systems". In: *International Journal of Production Research* 46.24 (2008), pp. 6841–6853.
- [35] Jan Fabian Ehmke, Stephan Meisel, and Dirk Christian Mattfeld. "Floating car based travel times for city logistics". In: *Transportation research part C: emerging technologies* 21.1 (2012), pp. 338–352.
- [36] Elsevier. *Scopus*. <http://www.scopus.com/home.url>. Visited on March 13th, 2015. 2014.
- [37] Xiang Fei, Chung-Cheng Lu, and Ke Liu. "A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction". In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1306–1318.
- [38] Ronald de Feijter, Joseph JM Evers, and Gabriel Lodewijks. "Improving travel-time reliability by the use of trip booking". In: *Intelligent Transportation Systems, IEEE Transactions on* 5.4 (2004), pp. 288–292.
- [39] Miguel Andres Figliozzi. "The impacts of congestion on commercial vehicle tour characteristics and costs". In: *Transportation research part E: logistics and transportation review* 46.4 (2010), pp. 496–506.
- [40] Miguel Andres Figliozzi. "The impacts of congestion on time-definitive urban freight distribution networks CO₂ emission levels: Results from a case study in Portland, Oregon". In: *Transportation Research Part C: Emerging Technologies* 19.5 (2011), pp. 766–778.
- [41] Jay W Forrester. "System dynamics, systems thinking, and soft OR". In: *System Dynamics Review* 10.2–3 (1994), pp. 245–256.
- [42] Anthony S Fowkes et al. "How highly does the freight transport industry value journey time reliability—and for what reasons?" In: *International Journal of Logistics: Research and Applications* 7.1 (2004), pp. 33–43.
- [43] Y. Freund and R.E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting". In: *Computational learning theory*. Springer. 1995, pp. 23–37.

- [44] Y. Freund and R.E. Schapire. “Experiments with a New Boosting Algorithms”. In: *Machine Learning: Proceedings of the Thirteenth International Conference*. 1996, pp. 148–156.
- [45] Timothy D Fry and Joan M Donohue. “Outlets for operations management research: a DEA assessment of journal quality and rankings”. In: *International Journal of Production Research* 51.23-24 (2013), pp. 7501–7526.
- [46] Barney G Glaser and Anselm L Strauss. *The discovery of grounded theory: Strategies for qualitative research*. Transaction Publishers, 1967.
- [47] Thomas F Golob and Amelia C Regan. “Impacts of highway congestion on freight operations: perceptions of trucking industry managers”. In: *Transportation Research Part A: Policy and Practice* 35.7 (2001), pp. 577–599.
- [48] Thomas F Golob and Amelia C Regan. “Trucking industry preferences for traveler information for drivers using wireless Internet-enabled devices”. In: *Transportation Research Part C: Emerging Technologies* 13.3 (2005), pp. 235–250.
- [49] Google Maps API Web Services. *The Google Directions API*. Mar. 13, 2015. URL: <https://developers.google.com/%20maps/documentation/directions/>.
- [50] Shirley Gregor and Alan R Hevner. “Positioning and Presenting Design Science Research for Maximum Impact”. In: *MIS Quarterly* 37.2 (2013), pp. 337–355.
- [51] Jifa Gu and Xijin Tang. “Meta-synthesis approach to complex system modeling”. In: *European Journal of Operational Research* 166.3 (2005), pp. 597–614.
- [52] Randolph W Hall. “Route choice and advanced traveler information systems on a capacitated and dynamic network”. In: *Transportation Research Part C: Emerging Technologies* 4.5 (1996), pp. 289–306.
- [53] Ray R Hashemi et al. “A neural network for transportation safety modeling”. In: *Expert Systems with Applications* 9.3 (1995), pp. 247–256.
- [54] David Hensher. *Institute of Transport and Logistics Studies’ Journal Rankings for Transport, Logistics and Supply Chain Management*. 2011. URL: http://rstrail.tudelft.nl/sites/default/files/journals%7B%5C_%7Dtransport-logistics-journal-rankings-110530%7B%5C_%7D3.pdf.

- [55] Aude Hofleitner, Ryan Herring, and Alexandre Bayen. “Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning”. In: *Transportation Research Part B: Methodological* 46.9 (2012), pp. 1097–1122.
- [56] Yaron Hollander and Ronghui Liu. “Estimation of the distribution of travel times by repeated simulation”. In: *Transportation Research Part C: Emerging Technologies* 16.2 (2008), pp. 212–231.
- [57] Takamichi Hosoda et al. “Is there a benefit to sharing market sales information? Linking theory and practice”. In: *Computers & industrial engineering* 54.2 (2008), pp. 315–326.
- [58] Kyle Hyndman, Santiago Kraiselburd, and Noel Watson. “Aligning capacity decisions in supply chains when demand forecasts are private information: Theory and experiment”. In: *Manufacturing & Service Operations Management* 15.1 (2013), pp. 102–117.
- [59] M.C. Jackson and P. Keys. “Towards a system of systems methodologies”. In: *The Journal of the Operations Research Society* 35.6 (1984), pp. 473–486.
- [60] V Jain, S Wadhwa, and SG Deshmukh. “Supplier selection using fuzzy association rules mining approach”. In: *International Journal of Production Research* 45.6 (2007), pp. 1323–1353.
- [61] Erik Jenelius. “The value of travel time variability with trip chains, flexible scheduling and correlated travel times”. In: *Transportation Research Part B: Methodological* 46.6 (2012), pp. 762–780.
- [62] Erik Jenelius, Lars-Göran Mattsson, and David Levinson. “Traveler delay costs and value of time with trip chains, flexible activity scheduling and information”. In: *Transportation Research Part B: Methodological* 45.5 (2011), pp. 789–807.
- [63] Hossein Julia, Maged Dessouky, and Petros A Ioannou. “Real-time estimation of travel times along the arcs and arrival times at the nodes of dynamic stochastic networks”. In: *Intelligent Transportation Systems, IEEE Transactions on* 9.1 (2008), pp. 97–110.
- [64] Payman Julia and Robert C Leachman. “Long-and short-run supply-chain optimization models for the allocation and congestion management of containerized imports from Asia to the United States”. In: *Transportation Research Part E: Logistics and Transportation Review* 47.5 (2011), pp. 593–608.

- [65] Yiannis Kamarianakis, H Oliver Gao, and Poulicos Prastacos. “Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions”. In: *Transportation Research Part C: Emerging Technologies* 18.5 (2010), pp. 821–840.
- [66] Seungmo Kang, Juan C Medina, and Yanfeng Ouyang. “Optimal operations of transportation fleet for unloading activities at container ports”. In: *Transportation Research Part B: Methodological* 42.10 (2008), pp. 970–984.
- [67] Roman Kapuscinski et al. “Inventory decisions in Dell’s supply chain”. In: *Interfaces* 34.3 (2004), pp. 191–205.
- [68] Wolfgang Ketter et al. “Detecting and forecasting economic regimes in multi-agent automated exchanges”. In: *Decision Support Systems* 47.4 (2009), pp. 307–318.
- [69] Wolfgang Ketter et al. “Real-time tactical and strategic sales management for intelligent agents guided by economic regimes”. In: *Information Systems Research* 23.4 (2012), pp. 1263–1283.
- [70] Akram Khaleghei Ghosheh Balagh, Farnoosh Naderkhani, and Viliam Makis. “Highway accident modeling and forecasting in winter”. In: *Transportation research part A: policy and practice* 59 (2014), pp. 384–396.
- [71] Abbas Khosravi et al. “A genetic algorithm-based method for improving quality of travel time prediction intervals”. In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1364–1376.
- [72] Abbas Khosravi et al. “Prediction intervals to account for uncertainties in travel time prediction”. In: *Intelligent Transportation Systems, IEEE Transactions on* 12.2 (2011), pp. 537–547.
- [73] Kap Hwan Kim, Keung Mo Lee, and Hark Hwang. “Sequencing delivery and receiving operations for yard cranes in port container terminals”. In: *International Journal of Production Economics* 84.3 (2003), pp. 283–292.
- [74] Kap Hwan Kim, Young-Man Park, and Mi-Ju Jin. “An optimal layout of container yards”. In: *OR Spectrum* 30.4 (2008), pp. 675–695.
- [75] Estelle RS Kone and Mark H Karwan. “Combining a new data classification technique and regression analysis to predict the Cost-To-Serve new customers”. In: *Computers & Industrial Engineering* 61.1 (2011), pp. 184–197.

- [76] Jukka Korpela and Markku Tuominen. "Inventory forecasting with a multiple criteria decision tool". In: *International journal of production economics* 45.1 (1996), pp. 159–168.
- [77] Panos Kouvelis, Charles L Munson, and Shilei Yang. "Robust Structural Equations for Designing and Monitoring Strategic International Facility Networks". In: *Production and Operations Management* 22.3 (2013), pp. 535–554.
- [78] William HK Lam, Hu Shao, and Agachai Sumalee. "Modeling impacts of adverse weather conditions on a road network with uncertainties in demand and supply". In: *Transportation research part B: methodological* 42.10 (2008), pp. 890–910.
- [79] Maurice Larrain. "The PMI, the T-Bill and Inventories: A Comparative Analysis of Neural Network and Regression Forecasts". In: *Journal of Supply Chain Management* 43.2 (2007), pp. 39–51.
- [80] Henry CW Lau, George TS Ho, and Yi Zhao. "A demand forecast model using a combination of surrogate data analysis and optimal neural network approach". In: *Decision Support Systems* 54.3 (2013), pp. 1404–1416.
- [81] Robert C Leachman and Payman Jula. "Congestion analysis of waterborne, containerized imports from Asia to the United States". In: *Transportation Research Part E: Logistics and Transportation Review* 47 (2011), pp. 992–1004.
- [82] Roger Lederman and Laura Wynter. "Real-time traffic estimation using data expansion". In: *Transportation Research Part B: Methodological* 45.7 (2011), pp. 1062–1079.
- [83] Hui Li et al. "Supply chain trust diagnosis (SCTD) using inductive case-based reasoning ensemble (ICBRE): The case of general competence trust diagnosis". In: *Applied Soft Computing* 12.8 (2012), pp. 2312–2321.
- [84] Li Li and Li Zhiheng. "Freeway travel-time estimation based on temporal-spatial queuing model". In: *Intelligent Transportation Systems, IEEE Transactions on* 14.3 (2013), pp. 1536–1541.
- [85] Ruimin Li and Geoffrey Rose. "Incorporating uncertainty into short-term travel time predictions". In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1006–1018.

- [86] Zheng Li, David A Hensher, and John M Rose. “Willingness to pay for travel time reliability in passenger transport: A review and some new empirical evidence”. In: *Transportation research part E: logistics and transportation review* 46.3 (2010), pp. 384–403.
- [87] Andy Liaw and Matthew Wiener. “Classification and Regression by randomForest”. In: *R News* 2.3 (2002), pp. 18–22. URL: <http://CRAN.R-project.org/doc/Rnews/>.
- [88] *TeXLive*. May 4, 2016. URL: <https://www.tug.org/texlive/>.
- [89] Hong K Lo, XW Luo, and Barbara WY Siu. “Degradable transport network: travel time budget of travelers with heterogeneous risk aversion”. In: *Transportation Research Part B: Methodological* 40.9 (2006), pp. 792–806.
- [90] Canonical Ltd. *About Ubuntu*. May 4, 2016. URL: <http://www.ubuntu.com/about/about-ubuntu>.
- [91] Chi-Jie Lu and Yen-Wen Wang. “Combining independent component analysis and growing hierarchical self-organizing maps with support vector regression in product demand forecasting”. In: *International Journal of Production Economics* 128.2 (2010), pp. 603–613.
- [92] David A Menachof et al. “An analysis of the value of supply chain management periodicals”. In: *International Journal of Physical Distribution & Logistics Management* 39.2 (2009), pp. 145–165.
- [93] Jack R Meredith, Michelle D Steward, and Bruce R Lewis. “Knowledge dissemination in operations management: Published perceptions versus academic reality”. In: *Omega* 39.4 (2011), pp. 435–446.
- [94] Alastair Milne and Neil Maiden. “Power and politics in requirements engineering: embracing the dark side?” In: *Requirements Engineering* 17.2 (2012), pp. 83–98.
- [95] Robert M Monczka et al. “Success Factors in Strategic Supplier Alliances: The Buying Company Perspective”. In: *Decision Sciences* 29.3 (1998), pp. 553–577.
- [96] Seongmin Moon, Andrew Simpson, and Christian Hicks. “The development of a classification model for predicting the performance of forecasting methods for naval spare parts demand”. In: *International Journal of Production Economics* 143.2 (2013), pp. 449–454.
- [97] Roland M. Mueller. “A Meta-model for Inferring Inter-theory Relationships of Causal Theories”. In: *Proceedings of the 48th Hawaii International Conference on Service Sciences*. Jan. 1, 2015.

- [98] Katta G Murty et al. "A decision support system for operations in a container terminal". In: *Decision Support Systems* 39.3 (2005), pp. 309–332.
- [99] Katta G Murty et al. "Hongkong International Terminals gains elastic capacity using a data-intensive decision-support system". In: *Interfaces* 35.1 (2005), pp. 61–75.
- [100] Nikolaos A Mylonopoulos and Vasilis Theoharakis. "On site: global perceptions of IS journals". In: *Communications of the ACM* 44.9 (2001), pp. 29–33.
- [101] Rajeev Namboothiri and Alan L Erera. "Planning local container drayage operations given a port access appointment system". In: *Transportation Research Part E: Logistics and Transportation Review* 44.2 (2008), pp. 185–202.
- [102] ManWo Ng and S Travis Waller. "A computationally efficient methodology to characterize travel time reliability using the fast Fourier transform". In: *Transportation Research Part B: Methodological* 44.10 (2010), pp. 1202–1219.
- [103] Yu Marco Nie and Xing Wu. "Shortest path problem considering on-time arrival probability". In: *Transportation Research Part B: Methodological* 43.6 (2009), pp. 597–613.
- [104] Adegoke Oke and Marek Szwejcowski. "The relationship between UK manufacturers' inventory levels and supply, internal and market factors". In: *International Journal of Production Economics* 93 (2005), pp. 151–160.
- [105] Darryn Paterson and Geoff Rose. "A recursive, cell processing model for predicting freeway travel times". In: *Transportation Research Part C: Emerging Technologies* 16.4 (2008), pp. 432–453.
- [106] Michael Pearson, Ron Masson, and Anthony Swain. "Process control in an agile supply chain network". In: *International Journal of Production Economics* 128.1 (2010), pp. 22–30.
- [107] Stefanie Peer, Carl C Koopmans, and Erik T Verhoef. "Prediction of travel time variability for cost-benefit analysis". In: *Transportation Research Part A: Policy and Practice* 46.1 (2012), pp. 79–90.
- [108] Ken Peffers et al. "A Design Science Research Methodology for Information Systems Research". In: *Journal of Management Information Systems* 24.3 (2008), pp. 45–77.

- [109] Matthew EH Petering. “Decision support for yard capacity, fleet composition, truck substitutability, and scalability issues at seaport container terminals”. In: *Transportation Research Part E: Logistics and Transportation Review* 47.1 (2011), pp. 85–103.
- [110] Matthew EH Petering. “Development and simulation analysis of real-time, dual-load yard truck control systems for seaport container transshipment terminals”. In: *OR spectrum* 32.3 (2010), pp. 633–661.
- [111] Matthew EH Petering et al. “Development and simulation analysis of real-time yard crane control systems for seaport container transshipment terminals”. In: *OR spectrum* 31.4 (2009), pp. 801–835.
- [112] Andrew M Pettigrew. “Longitudinal field research on change: Theory and practice”. In: *Organization science* 1.3 (1990), pp. 267–292.
- [113] Mai-Ha Phan and Kap Hwan Kim. “Negotiating truck arrival times among trucking companies and a container terminal”. In: *Transportation Research Part E: Logistics and Transportation Review* 75 (2015), pp. 132–144.
- [114] Portbase. Mar. 16, 2016. URL: <https://www.portbase.com/en/>.
- [115] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2015. URL: <https://www.R-project.org/>.
- [116] L Rabelo et al. “Using system dynamics, neural nets, and eigenvalues to analyse supply chain behaviour. A case study”. In: *International Journal of Production Research* 46.1 (2008), pp. 51–71.
- [117] Sampath Rajagopalan. “Impact of variety and distribution system characteristics on inventory levels at US retailers”. In: *Manufacturing & Service Operations Management* 15.2 (2013), pp. 191–204.
- [118] Piet Rietveld, FR Bruinsma, and Daniel J van Vuuren. “Coping with unreliability in public transport chains: A case study for Netherlands”. In: *Transportation Research Part A: Policy and Practice* 35.6 (2001), pp. 539–559.
- [119] Rijkswaterstaat. *MTR+: Traffic measurements for the Netherlands*. Mar. 13, 2015. URL: http://www.rijkswaterstaat.nl/zakelijk/efficient%5C_onderhoud%20/maandelijkse%5C_telpuntrapportages%20/technische%5C_informatie%5C_mtr.aspx.
- [120] Royal Dutch Meteorological Institute. *Hourly Weather data for the Netherlands*. Mar. 13, 2015. URL: <http://www.knmi.nl/klimatologie/uurgegevens/selectie.cgi>.

- [121] Thomas L Saaty. "Decision making with the analytic hierarchy process". In: *Int. J. Services Sciences* 1.1 (2008), pp. 83–98.
- [122] SCImago Journal & Country Rank. *Journal Rankings*. 2014. URL: <http://www.scimagojr.com/journalrank.php>.
- [123] Alexander Serenko and Michael Dohan. "Comparing the expert survey and citation impact journal ranking methods: Example from the field of Artificial Intelligence". In: *Journal of Informetrics* 5.4 (2011), pp. 629–648.
- [124] Bülent Sezen. "Relative effects of design, integration and information sharing on supply chain performance". In: *Supply Chain Management: An International Journal* 13.3 (2008), pp. 233–240.
- [125] Jihui-Biing Sheu and Stephen G Ritchie. "A new methodology for incident detection and characterization on surface streets". In: *Transportation Research Part C: Emerging Technologies* 6.5 (1998), pp. 315–335.
- [126] Galit Shmueli. "To explain or to predict?" In: *Statistical Science* (2010), pp. 289–310.
- [127] Galit Shmueli and Otto R Koppius. "Predictive analytics in information systems research". In: *MIS Quarterly* 35.3 (2011), pp. 553–572.
- [128] Axel Simroth and Henryk Zahle. "Travel time prediction using floating car data applied to logistics planning". In: *Intelligent Transportation Systems, IEEE Transactions on* 12.1 (2011), pp. 243–253.
- [129] *Spilter Digital Brainstorming*. 2015. URL: <http://www.spilter.nl/portfolio-item/brainstorm-technieken/>.
- [130] Sjoerd van der Spoel. "Methods for making predictions in cross-chain contexts". In: *Cross-Chain Collaboration in the Fast Moving Consumer Goods Supply Chain*. Ed. by A.G. de Kok, J. van Dalen, and J. van Hillegersberg. 2015.
- [131] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. "A Benchmark for Predicting Turnaround Time for Trucks at a Container Terminal". In: *Proceedings of the Big Data Interoperability for Enterprises Workshop*. Ed. by Maria Iacob and Marten van Sinderen. 2016.
- [132] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. "Predictive Analytics for Truck Arrival Time Estimation: A Field Study at a European Distribution Center". In: *International Journal of Production Research* (forthcoming) (2015).

- [133] Sjoerd van der Spoel, Chintan Amrit, and Jos van Hillegersberg. “The role of domain analysis in prediction instrument development”. In: *Proceedings of the Big Data Interoperability for Enterprises Workshop*. Ed. by Maria Iacob and Marten van Sinderen. 2016.
- [134] Sjoerd van der Spoel, Maurice van Keulen, and Chintan Amrit. “Process prediction in noisy data sets: a case study in a Dutch hospital”. In: *Data-Driven Process Discovery and Analysis*. Springer, 2013, pp. 60–83.
- [135] Robert Stahlbock and Stefan Voß. “Operations research at container terminals: a literature update”. In: *Or Spectrum* 30.1 (2008), pp. 1–52.
- [136] Anthony Stathopoulos and Matthew G Karlaftis. “A multivariate state space approach for urban traffic flow modeling and prediction”. In: *Transportation Research Part C: Emerging Technologies* 11.2 (2003), pp. 121–135.
- [137] Dirk Steenken, Stefan Voß, and Robert Stahlbock. “Container terminal operation and operations research-a classification and literature review”. In: *OR spectrum* 26.1 (2004), pp. 3–49.
- [138] Jeffrey S Stonebraker et al. “Impact factor as a metric to assess journals where OM research is published”. In: *Journal of Operations Management* 30.1 (2012), pp. 24–43.
- [139] *TeXstudio*. May 4, 2016. URL: <http://www.texstudio.org/>.
- [140] Jie Sun et al. “Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches”. In: *Knowledge-Based Systems* 57 (2014), pp. 41–56.
- [141] Riska Asriana Sutrisnowati, Hyerim Bae, and Minseok Song. “Bayesian network construction from event log for lateness analysis in port logistics”. In: *Computers & Industrial Engineering* (2014).
- [142] P.N. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. Addison-Wesley, 2006.
- [143] Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*. R package version 4.1-10. 2015. URL: <https://CRAN.R-project.org/package=rpart>.
- [144] Thomson Reuters. *Journal Citation Reports*. 2014. URL: <http://thomsonreuters.com/journal-citation-reports/>.
- [145] Huizhao Tu et al. “Modeling travel time reliability of freeways using risk assessment techniques”. In: *Transportation Research Part A: Policy and Practice* 46.10 (2012), pp. 1528–1540.

- [146] B Vahdani, M Zandieh, and V Roshanaei. "A hybrid multi-stage predictive model for supply chain network collapse recovery analysis: a practical framework for effective supply chain network continuity management". In: *International Journal of Production Research* 49.7 (2011), pp. 2035–2060.
- [147] J. Van Belle, P. Valckenaers, and D. Cattrysse. "Cross-docking: State of the art". In: *Omega* 40 (2012), pp. 827–846.
- [148] CP Van Hinsbergen, JWC Van Lint, and HJ Van Zuylen. "Bayesian committee of neural networks to predict travel times with confidence intervals". In: *Transportation Research Part C: Emerging Technologies* 17.5 (2009), pp. 498–509.
- [149] JWC Van Lint. "Online learning solutions for freeway travel time prediction". In: *Intelligent Transportation Systems, IEEE Transactions on* 9.1 (2008), pp. 38–47.
- [150] JWC Van Lint, SP Hoogendoorn, and Henk J van Zuylen. "Accurate freeway travel time prediction with state-space neural networks under missing data". In: *Transportation Research Part C: Emerging Technologies* 13.5 (2005), pp. 347–369.
- [151] JWC Van Lint, Henk J van Zuylen, and H Tu. "Travel time unreliability on freeways: why measures based on variance tell only half the story". In: *Transportation Research Part A: Policy and Practice* 42.1 (2008), pp. 258–277.
- [152] John Venable, Jan Pries-Heje, and Richard Baskerville. "A comprehensive framework for evaluation in design science research". In: *Design Science Research in Information Systems, Advances in Theory and Practice*. Springer, 2012, pp. 423–438.
- [153] Viswanath Venkatesh, Susan A. Brown, and Hillol Bala. "Bridging the qualitative-quantitative divide: Guidelines for conducting mixed methods research in information systems". In: *MIS Quarterly* 37.1 (2013), pp. 21–54.
- [154] T Warren Liao and Pei-Chann Chang. "Impacts of forecast, inventory policy, and lead time on supply chain inventory—a numerical study". In: *International Journal of Production Economics* 128.2 (2010), pp. 527–537.
- [155] Jane Webster and Richard T Watson. "Analyzing the past to prepare for the future: Writing a literature review". In: *MIS Quarterly* 26.2 (2002), p. 3.

- 156 Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. ISBN: 978-0-387-98140-6. URL: <http://ggplot2.org>.
- 157 Brent D Williams et al. "Predicting retailer orders with POS and order data: The inventory balance effect". In: *European Journal of Operational Research* 232.3 (2014), pp. 593–600.
- 158 Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression". In: *Intelligent Transportation Systems, IEEE Transactions on* 5.4 (2004), pp. 276–281.
- 159 Jei-Zheng Wu. "Inventory write-down prediction for semiconductor manufacturing considering inventory age, accounting principle, and product structure with real settings". In: *Computers & Industrial Engineering* 65.1 (2013), pp. 128–136.
- 160 S David Wu et al. "Managing short life-cycle technology products for Agere Systems". In: *Interfaces* 36.3 (2006), pp. 234–247.
- 161 Chengcheng Xu et al. "Predicting crash likelihood and severity on freeways with real-time loop detector data". In: *Accident Analysis & Prevention* 57 (2013), pp. 30–39.
- 162 Hai Yang. "Multiple equilibrium behaviors and advanced traveler information systems with endogenous market penetration". In: *Transportation Research Part B: Methodological* 32.3 (1998), pp. 205–218.
- 163 Jiyouon Yeon, Lily Elefteriadou, and Siriphong Lawphongpanich. "Travel time estimation on a freeway using Discrete Time Markov Chains". In: *Transportation Research Part B: Methodological* 42.4 (2008), pp. 325–338.
- 164 Mehmet Yildirimoglu and Nikolas Geroliminis. "Experienced travel time prediction for congested freeways". In: *Transportation Research Part B: Methodological* 53 (2013), pp. 45–63.
- 165 Bin Yu, William HK Lam, and Mei Lam Tam. "Bus arrival time prediction at bus stop with multiple routes". In: *Transportation Research Part C: Emerging Technologies* 19.6 (2011), pp. 1157–1170.
- 166 Rongjie Yu and Mohamed Abdel-Aty. "Utilizing support vector machine in real-time crash risk evaluation". In: *Accident Analysis & Prevention* 51 (2013), pp. 252–259.

- 167 Elisabeth Zehendner and Dominique Feillet. “Benefits of a truck appointment system on the service quality of inland transport modes at a multimodal container terminal”. In: *European Journal of Operational Research* 235.2 (2014), pp. 461–469.
- 168 Xianyuan Zhan et al. “Urban link travel time estimation using large-scale taxi data with partial information”. In: *Transportation Research Part C: Emerging Technologies* 33 (2013), pp. 37–49.
- 169 Chuqian Zhang et al. “Storage space allocation in container terminals”. In: *Transportation Research Part B: Methodological* 37.10 (2003), pp. 883–903.
- 170 Xiaoyan Zhang and John A Rice. “Short-term travel time prediction”. In: *Transportation Research Part C: Emerging Technologies* 11.3 (2003), pp. 187–210.
- 171 Wenjuan Zhao and Anne V Goodchild. “The impact of truck arrival information on container terminal rehandling”. In: *Transportation Research Part E: Logistics and Transportation Review* 46.3 (2010), pp. 327–343.

E Details of the domain analysis review

<i>Analysis type</i>	<i>Simple</i>	<i>Domain type Complex</i>
<i>Soft-exclusive</i>	2/47: 21, 68	41/47: 6, 9, 10, 13, 18, 24, 28, 30, 34, 40, 57, 58, 60, 64, 65, 67, 69, 70, 75–77, 79–81, 83, 91, 95, 96, 104, 106, 117, 124, 128, 132, 140, 146, 154, 157, 159, 161, 166
<i>Soft-inclusive</i>	0/47	4/47: 15, 53, 116, 160

Table E.1

Results of the literature review and domain complexity analysis

F Details of the arrival time review

Concepts													Methods			
Weather	Congestion/flow	Speed	Distance	Type of cargo	Type of truck	Time of day/week/month/year	Cumulative previous deviation	Accidents/incidents	Road work	Traffic signal failures	Road condition	Driving style	Empirical data	Simulated data	Factor validation	Reference
								x					x	N		52
	x							x					x	x	L	125
	x													x	L	162
						x							x		E	2
											x	x		x	N	8
						x							x		N	7
								x					x		L	118
	x												x		N	47
	x	x											x		L	136
	x	x											x		L	170
			x	x	x	x							x		L	42

Continued on next page...

Concepts												Methods				
Weather	Congestion/flow	Speed	Distance	Type of cargo	Type of truck	Time of day/week/month/year	Cumulative previous deviation	Accidents/incidents	Road work	Traffic signal failures	Road condition	Driving style	Empirical data	Simulated data	Factor validation	Reference
x	x	x				x		x					x		N	158
	x													x	L	38
x								x	x	x				x	L	31
	x	x												x	L	150
x	x						x	x					x		L	48
								x	x	x				x	L	89
x	x	x			x	x					x			x	N	56
	x	x											x	x	L	105
x	x					x					x		x		L	163
x	x					x		x	x					x	L	78
		x				x							x		E	151
		x				x								x	N	63
		x											x		L	149
		x											x		L	148
						x								x	L	103
x								x			x		x	x	L	86
x	x					x		x		x				x	L	23
		x				x		x						x	L	102
		x											x		L	39
		x											x		L	40
		x											x		L	165
		x											x		L	37
x		x			x		x	x					x	x	L	71
x	x					x							x		L	85
		x				x							x		L	82

Continued on next page...

Concepts												Methods				
Weather	Congestion/flow	Speed	Distance	Type of cargo	Type of truck	Time of day/week/month/year	Cumulative previous deviation	Accidents/incidents	Road work	Traffic signal failures	Road condition	Driving style	Empirical data	Simulated data	Factor validation	Reference
x	x					x		x		x			x	L		62
	x												x	L		128
x	x					x		x	x		x		x	L		72
	x					x							x	L		35
x	x					x			x					x	L	61
	x											x	x	L		55
x	x					x							x	L		107
	x							x					x	L		145
	x					x							x	L		168
x	x				x							x	x	L		4
	x												x	L		164
	x												x	N		17
	x												x	L		84
Mentions:																
15	38	7	1	1	4	20	2	15	5	4	5	5				

Table F.1

Matrix of concepts found in the arrival time review. Meaning of the abbreviations in the factor validation column: **L**iterature, **E**mpirical, **N**one

G Distribution center survey

Figure G.1 shows the English version of the survey presented to truckers arriving at the DC. In total, the survey was distributed in twelve languages: Dutch, English, French, Spanish, German, Russian, Bulgarian, Romanian, Polish, Slovenian, and Czech. Table G.1 shows how many respondents there were for each language, ranked from most to fewest.

<i>Language</i>	<i>Respondents</i>	<i>Language</i>	<i>Respondents</i>
Dutch	99	Russian	13
English	27	French	9
Polish	27	Spanish	8
German	21	Czech	7
Romanian	15	Bulgarian	4

Table G.1

Survey respondents per language. There were zero Slovenian responses.

Survey: Factors influencing arrival time

For scientific research, we are interested in which factors you think influence your arrival time at <redacted company name>. We will use your answers to better predict arrival times. All your answers will be treated confidentially and anonymously. Thanks in advance for filling out the survey.

1. Date:
2. What time did you arrive?

..... :
hour minute
3. What was your planned arrival time?

..... :
hour minute
4. What was your previous stop?
5. What is your next stop?
6. When do you plan to deliver the goods you picked up at <redacted company name> today?

..... :
hour minute
7. Where will you deliver the goods you picked up at <redacted company name> today?

.....

Please answer how each factor below influenced your arrival time.

	Made me arrive much earlier		No influence		Made me arrive much later
Weather	☐	☐	☐	☐	☐
Traffic/congestion	☐	☐	☐	☐	☐
My driving style	☐	☐	☐	☐	☐
Cargo	☐	☐	☐	☐	☐
My company's planning	☐	☐	☐	☐	☐
Distance I had to travel	☐	☐	☐	☐	☐
Type of truck /trailer	☐	☐	☐	☐	☐
Time of day	☐	☐	☐	☐	☐
Other:					
.....	☐	☐	☐	☐	☐
.....	☐	☐	☐	☐	☐

Figure G.1

The survey presented to truckers who arrived at the DC. <redacted company name> denotes a placeholder for the actual company name. This was done because the name of the company where this survey was conducted is confidential

H Distribution center inter variable correlations

	<i>tardiness</i>	<i>meanwind</i>	<i>maxgust</i>	<i>meantemp</i>	<i>mintemp</i>	<i>maxtemp</i>	<i>max.traffic1</i>	<i>min.traffic1</i>	<i>mean.traffic1</i>	<i>max.traffic2</i>
<i>tardiness</i>										
<i>meanwind</i>	-0.02									
<i>maxgust</i>	0	0.78								
<i>meantemp</i>	0.02	-0.13	0.01							
<i>mintemp</i>	0	0.09	0.15	0.92						
<i>maxtemp</i>	0.02	-0.24	-0.05	0.96	0.82					
<i>max.traffic1</i>	-0.2	0.04	0.02	-0.03	-0.01	-0.04				
<i>min.traffic1</i>	-0.2	0.06	0.02	0.04	0.06	0.01	0.81			
<i>mean.traffic1</i>	-0.2	0.03	0.01	-0.02	0.01	-0.03	0.98	0.88		
<i>max.traffic2</i>	-0.18	0.02	0	-0.03	-0.01	-0.04	0.84	0.83	0.87	
<i>min.traffic2</i>	-0.19	-0.01	-0.01	0.08	0.09	0.07	0.76	0.91	0.83	0.88
<i>mean.traffic2</i>	-0.19	0.02	0	-0.02	0	-0.03	0.85	0.87	0.88	0.98
<i>max.traffic3</i>	-0.17	0.02	0	0	0.01	-0.01	0.68	0.76	0.71	0.92
<i>min.traffic3</i>	-0.21	0.04	0.01	0.01	0.02	-0.01	0.58	0.84	0.66	0.81
<i>mean.traffic3</i>	-0.18	0.02	0.01	-0.01	0.01	-0.02	0.65	0.78	0.68	0.91
<i>mean_t..1</i>	-0.07	-0.04	0.04	0.89	0.84	0.83	0.22	0.41	0.26	0.29

	tardiness	meanwind	maxgust	meantemp	mintemp	maxtemp	max.traffic1	min.traffic1	mean.traffic1	max.traffic2
mean_w..1	-0.03	0.82	0.65	-0.05	0.14	-0.15	0.14	0.25	0.16	0.19
max_w..1	-0.03	0.8	0.66	-0.04	0.14	-0.15	0.17	0.28	0.19	0.22
precip_amount..1	0.02	0.13	0.13	-0.04	0.05	-0.07	0	-0.03	-0.01	0
fog..1	0.01	-0.22	-0.22	0.01	-0.03	0.07	-0.16	-0.22	-0.17	-0.2
rain..1	0.02	0.34	0.3	-0.07	0.08	-0.15	-0.01	-0.02	-0.01	-0.02
mean_t..2	-0.05	-0.01	0.05	0.9	0.86	0.84	0.15	0.38	0.2	0.24
mean_w..2	-0.03	0.82	0.61	-0.09	0.11	-0.2	0.13	0.28	0.16	0.2
max_w..2	-0.03	0.8	0.61	-0.08	0.12	-0.19	0.15	0.31	0.18	0.23
precip_amount..2	0.01	0.21	0.22	-0.02	0.08	-0.08	0	-0.02	0	0
fog..2	-0.02	-0.29	-0.23	0	-0.05	0.08	-0.11	-0.23	-0.13	-0.19
rain..2	0.03	0.36	0.33	-0.07	0.1	-0.17	0	-0.01	0	-0.01
mean_t..3	-0.07	-0.03	0.03	0.91	0.87	0.85	0.1	0.34	0.16	0.19
mean_w..3	-0.03	0.76	0.59	-0.08	0.06	-0.17	0.1	0.23	0.12	0.16
max_w..3	-0.04	0.76	0.61	-0.07	0.08	-0.16	0.1	0.26	0.13	0.17
precip_amount..3	0.01	0.27	0.33	0	0.09	-0.05	-0.01	0	0	-0.01
fog..3	-0.03	-0.24	-0.23	-0.02	-0.05	0.04	-0.05	-0.14	-0.07	-0.1

	min.traffic2	mean.traffic2	max.traffic3	min.traffic3	mean.traffic3
mean.traffic2	0.91				
max.traffic3	0.83	0.93			
min.traffic3	0.87	0.81	0.83		
mean.traffic3	0.84	0.92	0.98	0.88	
mean_t..1	0.38	0.3	0.3	0.37	0.31

Appendices

	<i>min.traffic2</i>	<i>mean.traffic2</i>	<i>max.traffic3</i>	<i>min.traffic3</i>	<i>mean.traffic3</i>	<i>mean_t..1</i>	<i>mean_w..1</i>	<i>max_w..1</i>	<i>precip_amount..1</i>	<i>fog..1</i>
mean_w..1	0.22	0.2	0.21	0.25	0.22	0.09				
max_w..1	0.26	0.23	0.23	0.28	0.25	0.12	0.99			
precip_amount..1	-0.02	0	0.01	-0.03	0	-0.03	0.17	0.19		
fog..1	-0.2	-0.2	-0.2	-0.21	-0.2	-0.13	-0.3	-0.3	-0.04	
rain..1	-0.01	-0.02	-0.01	-0.02	-0.02	-0.03	0.39	0.41	0.5	-0.15
mean_t..2	0.35	0.26	0.27	0.36	0.29	0.98	0.11	0.14	-0.02	-0.11
mean_w..2	0.26	0.21	0.22	0.3	0.24	0.05	0.9	0.9	0.11	-0.28
max_w..2	0.29	0.24	0.25	0.33	0.27	0.08	0.9	0.91	0.12	-0.29
precip_amount..2	-0.01	-0.01	-0.01	-0.02	-0.01	0	0.29	0.3	0.47	-0.08
fog..2	-0.21	-0.2	-0.22	-0.24	-0.22	-0.15	-0.34	-0.35	-0.07	0.66
rain..2	-0.01	-0.01	-0.01	-0.02	-0.01	-0.02	0.41	0.42	0.41	-0.14
mean_t..3	0.33	0.2	0.21	0.33	0.24	0.95	0.1	0.12	-0.02	-0.1
mean_w..3	0.21	0.16	0.18	0.27	0.2	0.02	0.7	0.7	-0.01	-0.25
max_w..3	0.23	0.18	0.19	0.3	0.21	0.05	0.73	0.73	0.01	-0.26
precip_amount..3	0.01	-0.01	-0.02	0	-0.02	0.04	0.35	0.37	0.17	-0.08
fog..3	-0.13	-0.11	-0.14	-0.15	-0.15	-0.08	-0.28	-0.29	-0.06	0.55

	<i>rain..1</i>	<i>mean_t..2</i>	<i>mean_w..2</i>	<i>max_w..2</i>	<i>precip_amount..2</i>	<i>fog..2</i>	<i>rain..2</i>	<i>mean_t..3</i>	<i>mean_w..3</i>	<i>max_w..3</i>	<i>precip_amount..3</i>
<i>mean_t..2</i>	0.01										
<i>mean_w..2</i>	0.36	0.09									
<i>max_w..2</i>	0.36	0.12	0.99								
<i>precip_amount..2</i>	0.45	0	0.21	0.24							
<i>fog..2</i>	-0.19	-0.16	-0.35	-0.36	-0.1						
<i>rain..2</i>	0.68	0	0.38	0.41	0.61	-0.21					
<i>mean_t..3</i>	0.01	0.97	0.08	0.1	0.01	-0.14	-0.01				
<i>mean_w..3</i>	0.27	0.05	0.84	0.83	0.11	-0.32	0.33	0.07			
<i>max_w..3</i>	0.29	0.08	0.86	0.86	0.14	-0.33	0.35	0.1	0.99		
<i>precip_amount..3</i>	0.27	0.05	0.35	0.37	0.4	-0.1	0.4	0.02	0.2	0.25	
<i>fog..3</i>	-0.15	-0.1	-0.29	-0.29	-0.09	0.64	-0.18	-0.14	-0.33	-0.34	-0.09

Table H.1

Inter-variable correlation (Pearson's r coefficients) in the data set. **Bold** values indicate those correlations are significant at the $\alpha = 0.05$ level. As the number of variables is large, the table is split into multiple parts. Explanation of the variable names: t is temperature; w is wind; $precip$ is precipitation of any kind; max is maximum; min is minimum; and the numbers 1..3 mean that the variable concerns only that zone.

I Details of the turnaround time review

Table I.1 shows all papers found in the structured literature review of turnaround time for trucks at a container terminal. The findings from this review are used in both Chapter 3 and Chapter 5.

We performed a structured literature review as suggested by Webster and Watson [155] to establish which aspects of a container terminal affect truck turnaround time. We used Scopus [36] as our search engine to find literature. In order to ensure the quality of the papers we limited our search to the top 20 papers of fields related to the supply chain, transport and predictive analytics/big data.

To establish what the top-20 journals are for these fields, we combined the Thomson Reuters Journal Citation Reports [144], the AIS Senior Scholar's Basket of Journals [5], the University of Sydney Institute of Transport and Logistics Studies' Journal Rankings for Transport, Logistics and Supply Chain Management [54], the SCImago Journal and Country Rank [122] as well as other author's rankings of journals [45, 92, 93, 100, 123, 138]. Combining these sources resulted in a total of 137 journals.

Query The query “*container AND (truck OR intermod*) AND (*load* OR handl* OR transf* OR turnaround) AND (forecast* OR predict* OR model) AND (time OR duration)*” resulted in 29 papers.

Round 1 - Evaluating abstracts In the first round of review, we read the abstract of each paper in the result of the query. The inclusion criteria we used were that the paper (i.) Mentions containers (ii.) Mentions terminals (iii.) Mentions trucks. After round 1, 14 of 29 papers remained.

Round 2 - Evaluating full papers In the second round of review, we analyzed the full text of the 14 remaining papers. The inclusion criteria were that the paper (i.) Discusses port terminals (ii.) Discusses truck transport (iii.) Discusses containers (iv.) Discusses a prediction model for turnaround time (v.) Discusses variables related to/predictors of turnaround time.

After round 2, only 8 out of the 14 papers remained.

Rounds 3 and 4 - Forward and backward search Using the initial list of eight papers, we performed backward (papers referenced in selected papers) and forward (papers referencing selected papers) search. This resulted in another 63 (backward search) + 101 (forward search) = 164 papers. These were reviewed using the same criteria discussed above. After reviewing abstracts, 19 papers remained. After reviewing the complete papers, that number was reduced to 11, bringing the total number of selected papers to 19.

Table I.1 shows the findings from the 19 selected papers. Each column indicates which variables are mentioned in the papers as predictors of turnaround time.

Table 1.1
Variables related to truck turnaround times mentioned in the selected papers from the literature review

J Domain complexity assessment

Below are nine questions based on Jackson and Keys' definition of domain complexity [59]. The left- and rightmost answer possibilities for each question are shown in italics.

- S1.) The domain represents a situation where people are trying to act together [22, p. 277]: *Disagree* ↔ *Agree*
- S2.) The domain involves politics, myths and meanings [22, p. 278]: *Disagree* ↔ *Agree*
- S3.) The domain is well-structured and there is consensus on what the problem is [22, p. 276]: *Disagree* ↔ *Agree*
- S4.) The domain is fully observable [59, p. 476]: *Disagree* ↔ *Agree*
- S5.) The domain is governed by well-known laws of behavior [59, p. 476]: *Disagree* ↔ *Agree*
- S6.) The domain's parts have their own goals [59, p. 476]: *Disagree* ↔ *Agree*
- S7.) The domain is [59, p. 476]: *Probabilistic* ↔ *Deterministic*
- S8.) The domain is open to its environment [59, p. 476]: *Disagree* ↔ *Agree*
- S9.) The domain is subject to behavioral influences [59, p. 476]: *Disagree* ↔ *Agree*

§J.1 Calculating domain complexity

To calculate domain complexity we take the average of the values of the answers. For the questions where a higher value would reflect lower complexity (like question S7 we take $6 - \{\text{answer}\}$ as the value of the answer. Questions such as S2 and S9 form a group, as both are about the influence of behavior/politics. For each group of questions, we take the average of the answers to the questions in that group.

$$\text{Domain Complexity} = \frac{S1 + \frac{(S2+S9)}{2} + (6 - S3) + (6 - S4) + (6 - \frac{(S5+S7)}{2}) + S8}{6}$$

K List of journals

Below is a list of all journals used in the reviews in Chapter 1 and Chapter 3. The references show which sources indicate that a journal is highly relevant to the field of predictive analytics, data mining, transport and the supply chain. For completeness, the references are:

- [5] Association for Information Systems. *Senior Scholars' Basket of Journals*. 2014. URL: <http://aisnet.org/?SeniorScholarBasket>.
- [45] Timothy D Fry and Joan M Donohue. "Outlets for operations management research: a DEA assessment of journal quality and rankings". In: *International Journal of Production Research* 51.23-24 (2013), pp. 7501–7526.
- [54] David Hensher. *Institute of Transport and Logistics Studies' Journal Rankings for Transport, Logistics and Supply Chain Management*. 2011. URL: http://rstrail.tudelft.nl/sites/default/files/journals%7B%5C_%7Dtransport-logistics-journal-rankings-110530%7B%5C_%7D3.pdf.
- [92] David A Menachof et al. "An analysis of the value of supply chain management periodicals". In: *International Journal of Physical Distribution & Logistics Management* 39.2 (2009), pp. 145–165.
- [93] Jack R Meredith, Michelle D Steward, and Bruce R Lewis. "Knowledge dissemination in operations management: Published perceptions versus academic reality". In: *Omega* 39.4 (2011), pp. 435–446.
- [100] Nikolaos A Mylonopoulos and Vasilis Theoharakis. "On site: global perceptions of IS journals". In: *Communications of the ACM* 44.9 (2001), pp. 29–33.

- [122] SCImago Journal & Country Rank. *Journal Rankings*. 2014. URL: <http://www.scimagojr.com/journalrank.php>.
- [123] Alexander Serenko and Michael Dohan. “Comparing the expert survey and citation impact journal ranking methods: Example from the field of Artificial Intelligence”. In: *Journal of Informetrics* 5.4 (2011), pp. 629–648.
- [138] Jeffrey S Stonebraker et al. “Impact factor as a metric to assess journals where OM research is published”. In: *Journal of Operations Management* 30.1 (2012), pp. 24–43.
- [144] Thomson Reuters. *Journal Citation Reports*. 2014. URL: <http://thomsonreuters.com/journal-citation-reports/>.

<i>Journal Title</i>	<i>References</i>
Academy of Management Annals	100, 144
Academy of Management Journal	100, 144
Academy of Management Perspectives	100, 144
Academy of Management Review	100, 144
Accident Analysis and Prevention	54
Accounting, Organizations and Society	122
ACM Transactions on Algorithms	100
ACM Transactions on Architecture and Code Optimization	100
ACM Transactions on Computational Logic	100
ACM Transactions on Computer Systems	100
ACM Transactions on Database Systems	100
ACM Transactions on Design Automation of Electronic Systems	100
ACM Transactions on Embedded Computing Systems	100
ACM Transactions on Information Systems	100
ACM Transactions on Knowledge Discovery from Data	100
ACM Transactions on Mathematical Software	100
ACM Transactions on Modeling and Computer Simulation	100
Administrative Science Quarterly	144
AI Magazine	123
Annals of Operations Research	45, 93, 138

Continued on next page . . .

<i>Journal Title</i>	<i>References</i>
Annual Review of Information Science and Technology	144
Applied Soft Computing	144
Archives of Computational Methods in Engineering	144
Artificial Intelligence	123, 144
Communications of the ACM	100
Communications of the Association for Information Systems	100
Computer-Aided Civil and Infrastructure Engineering	144
Computers & Industrial Engineering	45, 93, 122, 138, 144
Computers & Operations Research	45, 93, 122, 138, 144
Computers & Mathematics with Applications	144
Data Mining and Knowledge Discovery	144
Decision Sciences	45, 92, 93, 100, 122, 138
Decision Support Systems	100, 144
Enterprise Information Systems	144
Environment and Planning A	54
European Journal of Industrial Engineering	144
European Journal of Information Systems	5, 100, 144
European Journal of Operational Research	45, 92, 93, 122, 138, 144
Evolutionary Computation	123, 144
Expert Systems with Applications	144
Finance and Stochastics	122
Group & Organization Management	144
Harvard Business Review	45, 92, 93, 100, 144
IEEE Computational Intelligence Magazine	144
IEEE Computer	100
IEEE Intelligent Systems	123, 144
IEEE Transactions on Evolutionary Computation	100, 123, 144
IEEE Transactions on Fuzzy Systems	100, 123, 144
IEEE Transactions on Industrial Informatics	100, 144
IEEE Transactions on Information Theory	100, 144
<i>Continued on next page . . .</i>	

<i>Journal Title</i>	<i>References</i>
IEEE Transactions on Intelligent Transportation Systems	54, 100
IEEE Transactions on Knowledge and Data Engineering	100, 123, 144
IEEE Transactions on Neural Networks	100, 123
IEEE Transactions on Pattern Analysis and Machine Intelligence	100, 123, 144
IEEE Transactions on Services Computing	100, 122, 144
IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Reviews	100, 123, 144
IIE Transactions	93, 138, 144
Image and Vision Computing	144
Industrial Management & Data Systems	144
Industrial Marketing Management	92
Information Fusion	144
Information Sciences	122, 144
Information Systems Journal	5, 100
Information Systems Research	5, 100
Information Technology and Management	144
INFORMS Journal on Computing	122, 138
Integrated Computer-Aided Engineering	144
Interfaces	93, 138
International Entrepreneurship and Management Journal	144
International Journal of Logistics: Research & Applications	92
International Journal of Management Reviews	144
International Journal of Operations & Production Management	45, 92, 93, 138
International Journal of Physical Distribution & Logistics Management	92
International Journal of Production Economics	45, 93, 122, 138, 144
International Journal of Production Research	45, 93, 138, 144
International Journal of Simulation Modelling	144
Journal of the Association for Information Systems	5, 100
Journal of Artificial Intelligence Research	123

Continued on next page . . .

<i>Journal Title</i>	<i>References</i>
Journal of Business & Economic Statistics	122
Journal of Business Logistics	45, 54, 92, 138
Journal of Information Technology	5, 144
Journal of Informetrics	122
Journal of Machine Learning Research	123, 144
Journal of Management	144
Journal of Management Studies	144
Journal of Marketing	92, 144
Journal of Management Information Systems	5, 100
Journal of Multivariate Analysis	122
Journal of Operations Management	45, 54, 92, 93, 138, 144
Journal of Organizational Behavior	144
Journal of Product Innovation Management	144
Journal of Quality Technology	144
Journal of Scheduling	122, 138
Journal of Service Research	144
Journal of Statistical Software	144
Journal of Strategic Information Systems	5, 100
Journal of Supply Chain Management	54, 92, 93, 138, 144
Journal of the ACM	144
Journal of the American Society for Information Science and Technology	144
Journal of the Operations Research Society	45, 93, 138
Knowledge-Based Systems	144
Long Range Planning	144
Machine Learning	123
Management Decision	144
Management Science	45, 54, 92, 93, 100, 122, 138, 144
Mathematical Programming	122, 144
Mathematics of Operations Research	45, 122, 138
MIS Quarterly	5, 100, 144
Manufacturing & Service Operations Management	45, 138, 144
Naval Research Logistics	45, 93, 138
<i>Continued on next page . . .</i>	

<i>Journal Title</i>	<i>References</i>
Neural Networks	123, 144
Omega-International Journal of Management Science	45, 93, 122, 138, 144
Operations Research	45, 92, 93, 122, 138, 144
OR Spektrum	122
Optimization Letters	144
Organizational Research Methods	144
Organization Science	100, 144
Pattern Recognition	123, 144
Pattern Recognition Letters	123
Production and Inventory Management Journal	45, 93
Production and Operations Management	45, 54, 93, 122, 138
Regional Studies	54
Research in Engineering Design	144
Research Policy	144
Sloan Management Review	45, 92
Statistics Surveys	122
Strategic Management Journal	144
Supply Chain Management: An International Journal	54, 92
Supply Chain Management Review	92
Systems & Control Letters	144
Technovation	144
Transportation Journal	54
Transportation Research Part A: Policy and Practice	54, 122
Transportation Research Part B: Methodological	54, 122, 144
Transportation Research Part C: Emerging Technologies	54, 122
Transportation Research Part E: Logistics and Transportation Review	92, 122, 138, 144
Transportation Science	45, 122, 138, 144

Table K.1

Journals used for the reviews in Chapter 1 and Chapter 3

L Concise method handbook

These pages show a concise, minimal handbook for the use of the PID-CD method. Every step of the method described in Chapter 2 is shown here graphically. It shows in detail how each stage breaks up into steps, and what happens in each step. That way, a user of the PID-CD can see at a glance what the required inputs and expected outputs of each step are.

The handbook starts with an overview. Here, the letters *D* and *C* stand for *divergence* and *convergence*, respectively. These two terms are at the core of our method: the initial stage is about establishing divergent hypotheses, whereas each latter stage is all about convergence. First, the hypotheses are converged into a smaller set of ‘interesting’ hypotheses, next the resulting data selection and cleaning strategies are further converged through constraints, then predictive models are converged using the same constraints, until finally only one remains and is made into a prediction instrument.

Each step in stages I and II shows how many ‘variants’ of hypotheses, strategies or models are still left after convergence/reduction by constraints. A variant is an instance of one these types, that could be unique, or a complete copy of another instance, and everything in between.

Overview

Stage I

Appendices

M Design & layout

This thesis was typeset using $\text{\TeX}$ Live [88], a distribution of $\text{\LaTeX}$ 2 ϵ , a free and open-source document preparation system. The typesetting engine used is $\text{\XeLaTeX}$, which facilitates the use of fonts other than those available in ‘standard’ $\text{\LaTeX}$. Text editing was done in $\text{\TeX}$ studio [139], on Ubuntu 14.04 [90], both available as free software.

Figures and plots Most images in this thesis were developed in $\text{\TikZ}$, a vector graphics language based on $\text{\PGF}$, which is included in $\text{\TeX}$ Live. The plots were made using $\text{\ggplot2}$ [156], a library from the R suite for statistical programming [115].

Bibliography $\text{\BibLaTeX}$ is the bibliography manager used for this thesis, with the `biber` backend. $\text{\BibLaTeX}$ is a reimplementaion of $\text{\BibTeX}$.

Fonts and titles The serif typeface used in the thesis is *Georgia*, designed by Matthew Carter, used in ‘normal’ weight. The sans-serif typeface used is *Lato*, designed by Łukasz Dziejczak, used in ‘light’ weight. The chapter and section titles were typeset using the `titlesec` package.

Packages Below is a complete list of the packages used for typesetting this thesis, each of which is documented online at <http://www.ctan.org/macros/contrib/>.

<code>adjustbox</code>	<code>amsmath</code>	<code>babel</code>
<code>afterpage</code>	<code>appendix</code>	<code>biblatex</code>
<code>algorithm2e</code>	<code>array</code>	<code>booktabs</code>

<code>caption</code>	<code>microtype</code>	<code>subcaption</code>
<code>etoolbox</code>	<code>mdframed</code>	<code>tabu</code>
<code>fancyhdr</code>	<code>multicols</code>	<code>tabularx</code>
<code>fontspec</code>	<code>multirow</code>	<code>textcase</code>
<code>forloop</code>	<code>nowidow</code>	<code>tikz</code>
<code>geometry</code>	<code>paralist</code>	<code>titlesec</code>
<code>graphicx</code>	<code>pgfplots</code>	<code>titletoc</code>
<code>hologo</code>	<code>relsize</code>	<code>tocloft</code>
<code>hyperref</code>	<code>rotating</code>	<code>totcount</code>
<code>longtable</code>	<code>setspace</code>	<code>xcolor</code>
<code>lscape</code>	<code>siunitx</code>	<code>xspace</code>
<code>makeidx</code>	<code>stmaryrd</code>	

§M.1 About the cover

The cover shows the design of a blueprint for prediction instrument development. Four steps of the PID-CD method are shown: first, at the top left, the domain is analyzed to produce hypotheses. Next, at the top right, these are used to clean and select data. Then, at the bottom left, these are processed into predictive models. Finally, at the bottom right, one predictive model is selected and turned into a prediction instrument.

About the author

Sjoerd van der Spoel was born on November 7th, 1986 in Almelo, The Netherlands. Although staying close to ‘home’ was never a requirement, he ended up choosing to study Business Information Technology at the University of Twente, as there was no programme offered anywhere in the Netherlands that was a better fit.

Since starting his Bachelor in 2004, Sjoerd has been actively involved in the vibrant student life in Enschede. Amongst others, he was the chairman of the board at study association Inter-Actief, a founder and first chairman of study association Inbitween, and a member of the board of the association of study associations. For the last two years, he was the chairman of the alumni association for Business Information Technology, Computer Science, and Human Media Interaction.

He finished his Master’s Degree in Business Information Technology in 2012 with a thesis on predicting duration and costs of medical treatments. Since starting his PhD in that same year, he has been involved in multiple projects, of which ‘4C4More’, a project funded by the Dutch Institute for Advanced Logistics, was the most important. For those projects, he conducted research at logistics and transport companies throughout the Netherlands, of which a selection is represented in the case studies in this thesis.