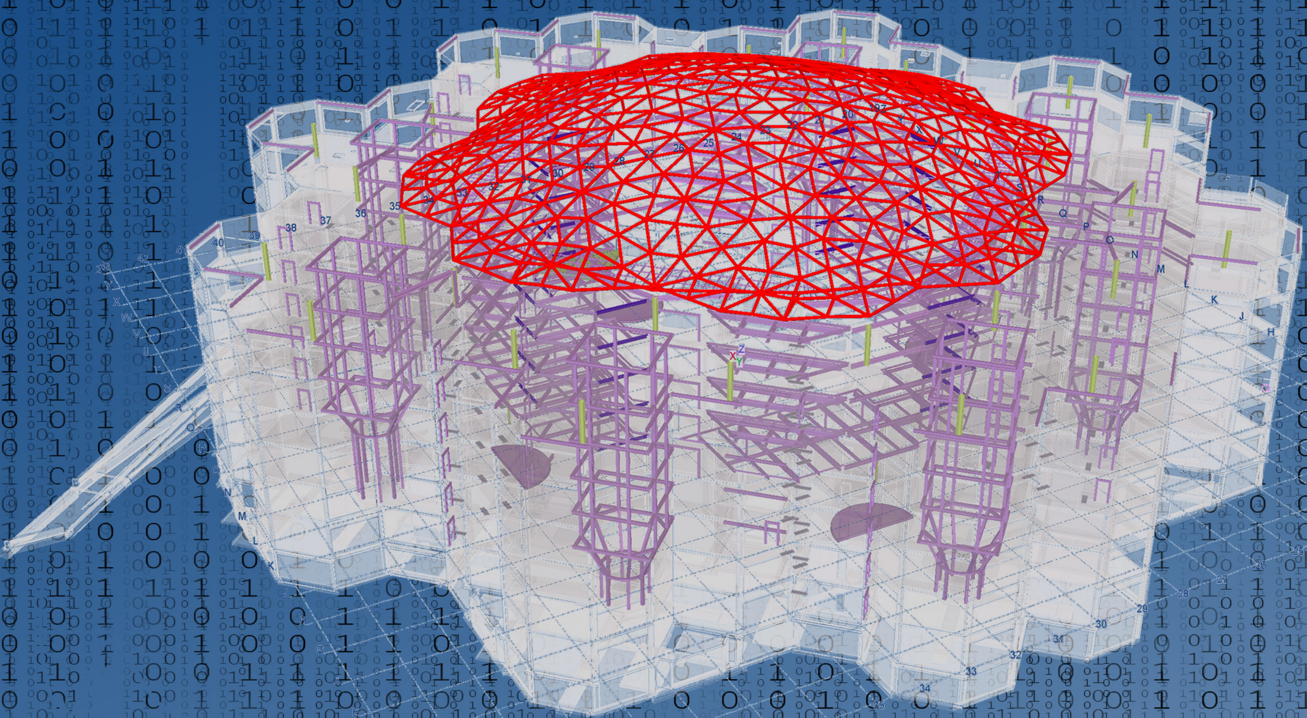
Predicting Manufacturing Times Using Building Information Models

L.A. Van der Plas



TU Delft Delft University of Technology

OOSTINGH ASK ROMEIN

Predicting manufacturing times using Building Information Models

A case study at Oostingh Staalbouw

by

L.A. van der Plas

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday October 29, 2019 at 14:00.

Student number:	4296702	
Project number:	2019.TEL.8375	
Project duration:	February 18, 2019 – October 29, 2019	
Thesis committee:	Prof. dr. R. R. Negenborn,	TU Delft, Committee chair
	Dr. ir. X. Jiang,	TU Delft, supervisor
	C. Oudshoorn	Company supervisor
	Dr. J. Vleugel	TU Delft, committee member

TU Delft

Delft University of Technology

FACULTY MECHANICAL, MARITIME AND MATERIALS ENGINEERING
Department Marine and Transport Technology

Mekelweg 2
2628 CD Delft
the Netherlands
Phone +31 (0)15-2782889
Fax +31 (0)15-2781397
www.mtt.tudelft.nl

Specialization:	Transport Engineering and Logistics
Report number:	2019.TEL.8375
Title:	**Predicting Manufacturing Times using Building Information Models**
Author:	L.A. van der Plas

Title (in Dutch)	Voorspellen van productietijden aan de hand van Bouw Informatie Modellen

Assignment:	Masters thesis
Confidential:	no
Initiator (university):	prof. dr. R.R. Negenborn
Initiator (company):	C. Oudshoorn
Supervisor:	dr. ir. X. Jiang
Date:	October 15, 2019

Preface

This report is written as graduation thesis for the master Mechanical Engineering track Transport Engineering and Logistics at the Delft University of Technology.

I would like to thank dr. ir. X. Jiang and prof. dr. R.R. Negenborn for being respectively the daily supervisor and the chair of the graduation committee for this thesis.

My gratitude goes out to Kees for giving me the opportunity to conduct this thesis at Oostingh Staalbouw. Your never ending enthusiasm and guidance during this project really kept me going day by day.

Last but not least, I want to thank my friends and family for all the support during my study. In all highs and lows in the past six years, you were there for me. Special thanks goes to my girlfriend, I wouldn't have managed without your support.

L.A. van der Plas
Katwijk, October 2019

Summary

In the Engineered-To-Order (**ETO**) branch of the construction industry, each construction project is unique. To realize these unique constructions, unique (steel) structural elements (hereafter referred to as 'products') are required. Due to the high complexity, low volume characteristics of these products the prediction of required manufacturing times per manufacturing step is not straight forward.

Currently the construction industry relies on experience based manufacturing time predictions of shop managers. This experience based approach is prone to estimation errors, leading to ineffective manufacturing schedules.

In the past decade, the construction industry started using Building Information Models (**BIM**) more frequently. In these models, information about the construction (physical properties of products) is stored. Simultaneously, increasing research on the application of data analysis for improving different aspects of the manufacturing process can be identified.

In this graduation thesis, a general manufacturing time prediction model is proposed using data from **BIM** and the manufacturing process. The validation of the proposed manufacturing time prediction model is performed in collaboration with Oostingh Staalbouw. Oostingh Staalbouw is a leading company specialized in the design, manufacturing and assembling of unique steel structures, located in the Netherlands. Currently, the prediction of manufacturing times results in a Mean Absolute Percentage Error (**MAPE**) of 0.60 (60%) at Oostingh Staalbouw. Based on the level of human related uncertainty in the data, the objective is set to reduce this **MAPE** to 0.30.

The manufacturing process of Oostingh Staalbouw consists of preprocessing, assembly, welding and coating. This research focused on the prediction of assembly and welding times. The available data for this research consists of physical properties (both quantitative and categorical) and realized manufacturing times per manufacturing step. A Spearman's correlation rank analysis showed that a monotonic relationship exists between the quantitative physical properties and realized manufacturing times. This analysis, however, remained ambiguous whether the relationships are linear or nonlinear.

Several different prediction models are identified after reviewing related literature. These prediction models are compared based on generality (able to deal with both linear and nonlinear relationships), ability to incorporate both quantitative and categorical input variables, robustness to uncertainty in the data and the complexity of the prediction model. From this comparison, an opportunity for a general prediction model is identified by combining a Support Vector Regression (**SVR**) and Linear Model Tree (**LMT**).

The proposed prediction model is able to split nonlinear relationships in several linear relationships using a decision tree. Using a greedy top down approach, the prediction model splits a node in a left and right child node. A linear model is build in both left and right child. The combination of split variable and corresponding split value leading to the biggest reduction in standard deviation of the residuals from the linear models is chosen. Upon splitting the relationships, an **SVR** prediction model is implemented in the nodes of the decision tree. Therefore the proposed prediction model is called the Support Vector Regression Model Tree (**SVRMT**). After the **SVRMT** prediction model is built, the tree is pruned in order to avoid overfitting. Sharp discontinuities between adjacent leaves are compensated through a smoothing procedure.

Testing of the proposed **SVRMT** prediction model is conducted in two phases. First the assumptions that the model is able to predict both linear and nonlinear relationships with added noise is verified. Afterwards, the **SVRMT** model is tested in three real case scenarios in the second, validation phase. The first scenario corresponds to the start of a new construction project where historical data from realized projects is used to train the prediction model. The second scenario corresponds to a significant part of the construction project having already been manufactured. The data from the manufactured products of the particular construction

project is used to train the model. At last, in the third scenario, the available data from the first and second scenario is combined to train the model.

The results of the second experimental phase showed a significant increase in prediction accuracy compared to the currently used experience based predictions. For the first scenario, projects differing significantly from historical projects are predicted less accurately than projects similar to historical projects. The second scenario turned out to be especially interesting for projects with a significant number of repetitive products. In the third scenario, outliers (both postive and negative) are flattened out, resulting in most consistent predictions in terms of absolute relative errors. The average **MAPE** of scenario 1, scenario 2 and scenario 3 are respectively 0.41, 0.38 and 0.38.

Overall, the results of this research show that predicting manufacturing times can be improved through implementation of prediction models. These prediction models are trained using historical data from **BIM** and manufacturing times. The proposed prediction model turned out to perform best in terms of prediction accuracy. There, however, still remains room for improving the prediction accuracy.

Samenvatting

In de Engineered-To-Order (**ETO**) branche van de bouw industrie is ieder gebouw uniek. Voor deze gebouwen zijn unieke structurele elementen ('producten') nodig. Door de hoge complexiteit en het lage volume van de producten is het voorspellen van productietijden per productiestap niet vanzelfsprekend.

Momenteel maakt de bouwindustrie gebruik van ervaring van productiemanagers om voorspellingen te doen over productietijden per productiestap. Aangezien deze methode gevoelig is voor fouten, leidt deze methode tot minder effectieve productieplanningen.

In het afgelopen decennium wordt met toenemende mate gebruik gemaakt van Bouw Informatie Modellen (**BIM**). Het voordeel van deze modellen is dat informatie over constructies (op product basis) kan worden opgeslagen. Tegelijkertijd wordt met toenemende mate onderzoek gedaan naar de implementatie van data analyse ter verbetering van productieprocessen.

In dit afstudeeronderzoek is een algemeen model voor het voorspellen van productietijden per productiestap voorgesteld. Dit voorspellingsmodel maakt gebruik van opgeslagen data in **BIM** en productietijden van geproduceerde producten. Het valideren van dit voorspellingsmodel is uitgevoerd in samenwerking met Oostingh Staalbouw. Oostingh Staalbouw is gespecialiseerd in het ontwerpen, produceren en assembleren van stalen constructies. Momenteel is de Mean Absolute Percentage Error (**MAPE**) van voorspelde productietijden bij Oostingh Staalbouw 0.60 (60%). Gebaseerd op (mensgerelateerde) onzekerheid in de data is het doel gesteld om deze **MAPE** te reduceren tot 0.30.

Het productieproces van Oostingh Staalbouw bestaat uit voorbewerking, aanbouw, las en coating stappen. Dit onderzoek is gefocust op het voorspellen van aanbouw en lastijden per product. Aanwezige data voor dit onderzoek bestaat uit fysieke eigenschappen (zowel kwantitatief als categorisch) en gerealiseerde productietijden per productiestap. Door middel van Spearman's correlatie rank kan een monotone relatie worden gevonden tussen kwantitatieve fysieke eigenschappen en gerealiseerde productietijden per productiestap. Deze analyse blijft echter ondubbelzinnig of de gevonden relatie lineair of niet lineair is.

Door middel van een literatuurstudie kunnen verschillende relevante voorspellingsmodellen worden geïdentificeerd. Deze modellen zijn vergeleken aan de hand van geïdentificeerde karakteristieken in de data. Het voorspellingsmodel moet voor zowel lineaire als niet lineaire relaties accurate voorspellingen geven. Het model moet robuust zijn tegen onzekerheid in de data en moet zowel kwantitatieve en categorische input variabelen kunnen verwerken. Tot slot moet het model inzichtelijk zijn om de acceptatiegraad binnen de conservatieve bouwindustrie te verhogen. Door middel van deze vergelijking is een nieuw voorspellingsmodel voorgesteld. Dit voorgestelde voorspellingsmodel is een combinatie tussen Support Vector Regression (**SVR**) en een Linear Model Tree (**LMT**). Het voorgestelde voorspellingsmodel wordt voor de rest van dit onderzoek de Support Vector Regression Model Tree (**SVRMT**) genoemd.

De **SVRMT** wordt opgebouwd door de relatie tussen input en output op te splitsen in lineaire segmenten. Voor ieder knooppunt wordt gezocht naar een optimale split variabele en bijbehorende split waarde. Door middel van een gretige zoekmethode, wordt de Model Tree van bovenaf gesplitst in een linker en rechter knoop. In beide knooppunten wordt een lineair model geplaatst. De combinatie die tot grootste reductie in standaarddeviatie van afwijking tussen lineair model en datapunten in de knoop leidt wordt gekozen als beste split. Nadat de relatie is opgesplitst in lineaire segmenten worden **SVR** modellen in alle knopen van de Model Tree geplaatst. Vervolgens wordt overfitting van de Model Tree tegengegaan door een 'pruning' methode. Tot slot worden grote verschillen tussen aangrenzende knopen gecompenseerd door middel van een 'smoothing' procedure.

Het voorgestelde **SVRMT** voorspellingsmodel is getest in twee fases. In de eerste fase zijn de aannames voor het **SVRMT** voorspellingsmodel geverifieerd. Vervolgens is het **SVRMT** voorspellingsmodel gevalideerd

door middel van drie verschillende case scenario's, waar gebruik wordt gemaakt van data van Oostingh Staalbouw. Het eerste scenario komt overeen met de start van het produceren van een nieuwe constructie. Op dit moment is geen gerealiseerde data aanwezig van de constructie. Daarom is het voorspellingsmodel volledig afhankelijk van data van historische projecten. In het tweede scenario wordt uitgegaan van het moment dat een significant deel van de constructie geproduceerd is. De gerealiseerde data van het project wordt gebruikt om het voorspellingsmodel te trainen. Tot slot wordt in het derde scenario gebruik gemaakt van zowel data van gerealiseerde projecten, als een significant geproduceerd deel van de te produceren constructie.

De resultaten van de tweede experimentele fase tonen een significante verbetering ten opzichte van de huidige voorspellingsmethode. In het eerste scenario blijkt de voorspellingsnauwkeurigheid afhankelijk van gelijkenissen tussen de verschillende constructies. Als een constructie significant verschillend is van de constructies die gebruikt zijn voor het trainen van het voorspellingsmodel, resulteert dit in minder nauwkeurige voorspellingen. Het tweede scenario blijkt met name interessant voor projecten met veel repeterende producten binnen de constructie. Scenario 3 resulteert in de meest constante resultaten tussen de verschillende constructies. De gemiddelde **MAPE** voor scenario 1, scenario 2 en scenario 3 zijn respectievelijk 0.41, 0.38 en 0.38.

De resultaten van dit onderzoek tonen aan dat de implementatie van voorspellingsmodellen leidt tot een significant verbeterde nauwkeurigheid in voorspelde productietijden. Het voorgestelede **SVRMT** voorspellingsmodel leidde tot meest nauwkeurige voorspelde productietijden per productiestap. De doelstelling van een **MAPE** van 0.30 is echter niet gehaald, wat betekent dat er ruimte voor verbetering blijft voor verder onderzoek.

Contents

List of Figures

List of Tables

Glossary

BIM	Building Information Modeling. vii–ix, 3–5, 7–10, 14, 18–20, 27–29, 46, 53, 54, 56
CNC	Computer numerical control. 11, 12
ETO	Engineered-to-order. ix, 2, 5, 11, 20, 54
LMT	Linear Model Tree. vii, ix, 25, 27–31, 33, 35, 39, 40, 42, 43, 51, 54
MAPE	Mean Absolute Percentage Error. vii–x, 2, 4, 16–18, 37–39, 43–46, 48–56
MeAPE	Median Absolute Percentage Error. 16, 49
MLR	Multiple linear regression. 3, 9, 20–28, 30, 31, 33–35, 39, 40, 42, 43, 51, 54
NN	Neural Network. 20, 21, 26, 28, 54
OLS	Ordinary Least Squares. 22, 24
RMSE	Root Mean Square Error. 36, 37
SVR	Support Vector Regression. vii, ix, 20, 21, 26–29, 31–35, 39, 42, 43, 49–51, 54, 55
SVRMT	Support Vector Regression Model Tree. vii, ix, x, xiv, 28–30, 33–35, 38–44, 46, 49–52, 54–56
TBR	Tree Based Regression. 20, 21, 24, 25, 27, 28, 33, 35, 51, 54

1 Introduction

1.1. Company background

Oostingh Staalbouw, since 2017 part of ASK Romein, is a leading company in designing, manufacturing and assembling unique and complex steel structures in the Netherlands. In the past years, the number of orders has increased rapidly. To cope with this development, improvements in the manufacturing process are required.

The uniqueness of the construction projects carried out by Oostingh Staalbouw, however, makes standardization a difficult task. In addition, structural elements (hereafter referred to as "products") are difficult to handle due to their size and length, making transport of products inconvenient. An example of a product is provided in Figure 1.1.

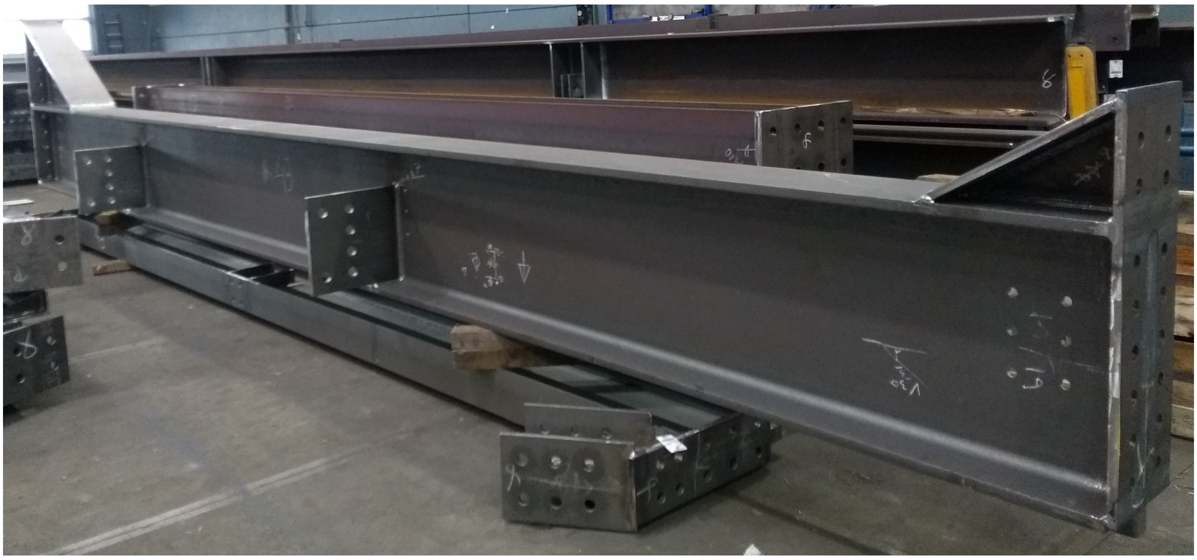


Figure 1.1: Example of a product manufactured at Oostingh Staalbouw

An example of a project being built by Oostingh Staalbouw in 2019 is the dome shown in Figure 1.2. This dome spans an area with a diameter of 70 meters and is built on an existing construction. The construction of this project requires a unique design, consisting of unique products.

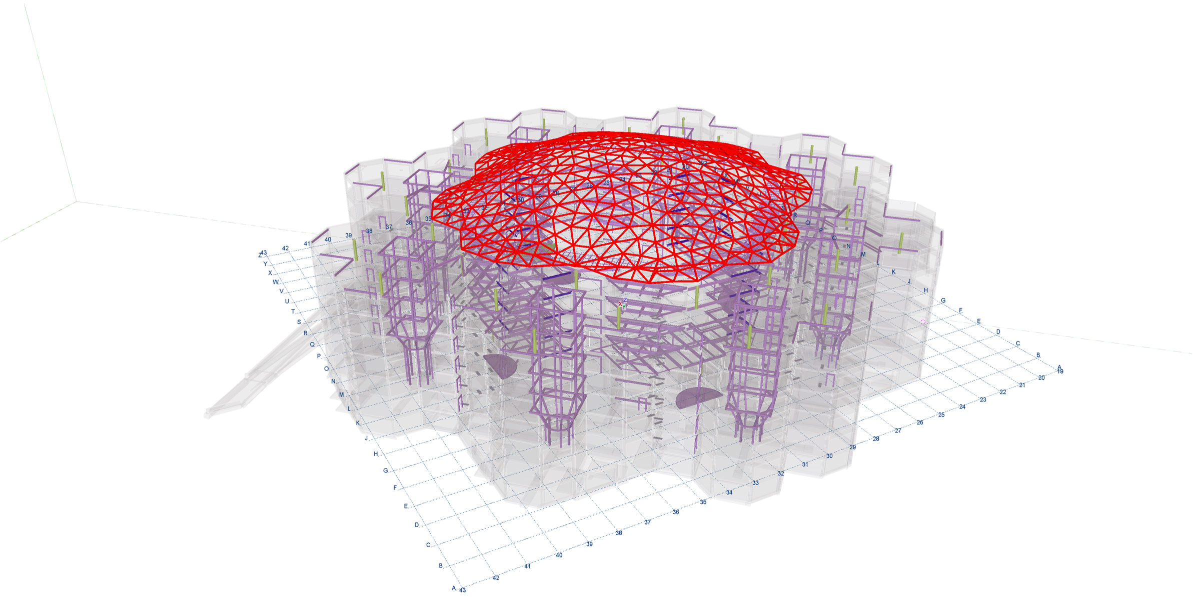


Figure 1.2: Example of project carried out by Oostingh Staalbouw

1.2. Problem definition

Due to the Engineered-to-order (**ETO**) nature of unique construction projects, each project demands different products, which have varying physical properties. This variation in physical properties results in significant variations in manufacturing time per product. A histogram of the variation in required welding time of different products at Oostingh Staalbouw is shown in Figure 1.3. Due to the low-volume, high complexity nature of these structural elements, it is challenging to predict manufacturing times accurately [2]. Unreliable predictions lead to difficulties in formulating manufacturing schedules which balance the production line and satisfy on-site demands [39].

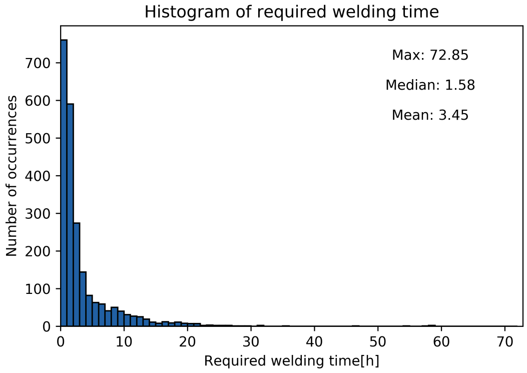
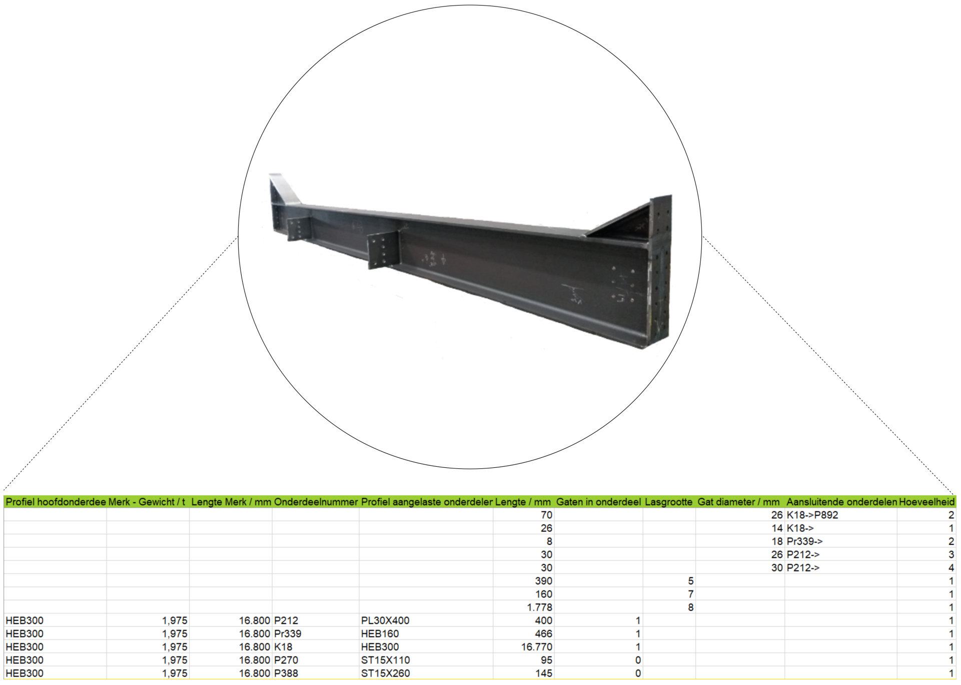


Figure 1.3: Histogram of required welding time per product for the period November 2018 - February 2019 at Oostingh Staalbouw

Currently, scheduling for off-site steel structure manufacturing processes is based on the experience of fabrication shop managers. These managers estimate the complexity of a group of products (part of a construction) on man-hours per tonne basis and manually produce a manufacturing schedule. This approach relies on experience and knowledge of the estimator and is prone to errors, leading to inaccurate manufacturing schedules [26]. At Oostingh Staalbouw, scheduling is currently conducted likewise, resulting in ineffective manufacturing schedules. Experience based predictions currently yield Mean Absolute Percentage Errors (**MAPE**) up to 0.60 (60%) at Oostingh Staalbouw.



Profiel hoofdonderdee	Merk - Gewicht / t	Lengte Merk / mm	Onderdeelnummer	Profiel aangelaste onderdeler	Lengte / mm	Gaten in onderdeel	Lasgrootte	Gat diameter / mm	Aansluitende onderdelen	Hoeveelheid
					70			26	K18->P892	2
					26			14	K18->	1
					8			18	Pr339->	2
					30			26	P212->	3
					30			30	P212->	4
					390		5			1
					160		7			1
					1.778		8			1
HEB300	1,975	16.800	P212	PL30X400	400	1				1
HEB300	1,975	16.800	Pr339	HEB160	466	1				1
HEB300	1,975	16.800	K18	HEB300	16.770	1				1
HEB300	1,975	16.800	P270	ST15X110	95	0				1
HEB300	1,975	16.800	P388	ST15X260	145	0				1

Figure 1.4: Visualization of data coupled to a product in BIM

Due to these prediction errors, and subsequently inaccurate manufacturing schedules, the workload of successive manufacturing steps becomes unbalanced. This results in disrupted product flows between the manufacturing steps, emerging in buffers of materials between the manufacturing steps. This leads to extra transport, waiting times and searching for materials.

In the past decade, construction engineers started using Building Information Models(**BIM**) to design complex structures. The main advantage of **BIM** is the integration of structural information with specific elements. This integration of information enhances interoperability in building projects significantly [52]. An example of information integration in **BIM** is provided in Figure 1.4. Despite the arsenal of possibilities, **BIM** based models are still mainly used for 3D modelling [26].

Meanwhile, data analysis is becoming an indispensable technique for improving manufacturing processes [11]. Data analysis can be implemented in numerous aspects of manufacturing, e.g. overall production management, planning and scheduling, quality monitoring and fault detection [11].

An opportunity arises based on above mentioned developments. Combining historical data from both **BIM** and the manufacturing process to improve the accuracy of predicted manufacturing times would therefore be an interesting research area [26]. This increased accuracy of predicted manufacturing times can be used to create more effective manufacturing schedules.

Research on predicting manufacturing times of structural elements based on historical data is, however, limited. Hu et al. [26] extracted physical properties, like size, weight and weld length of products from **BIM**. This extracted data is used in combination with historical data of manufacturing times from the manufacturing process to predict manufacturing times more accurately. A Multiple Linear Regression method (**MLR**) is compared with traditional, experience based predictions. The **MLR** method significantly outperformed the traditional approach in terms of prediction accuracy.

The study, however, is limited to the prediction of manufacturing times for entire projects, which is unsuitable for generating detailed manufacturing schedules. Therefore the aim of this graduation thesis is to predict manufacturing times per product per manufacturing step, paving the way for the generation of effective manufacturing schedules. Furthermore, the study from Hu et al. [26] only took into account **MLR** as prediction model, which is limited to processes where physical properties of products and manufacturing times have linear relationships. A general model for predicting manufacturing times using information from **BIM** is, to the best of the author's knowledge, still missing.

1.3. Objective

This study aims to propose a manufacturing time prediction model based on historical information stored in **BIM** and manufacturing times. Considering the wide variety of manufacturing processes in the construction industry, generality will be taken into account for the manufacturing time prediction model.

The proposed general manufacturing time prediction model will be validated in a case study. For this case study, the aim is to increase the accuracy of predictions of new, unique products manufactured at Oostingh Staalbouw. Currently, the Mean Absolute Percentage Error (**MAPE**) of the experience based prediction approach at Oostingh Staalbouw is up to 0.60 (60%). Uncertainty in the data accounts for an average relative prediction error of 0.25. The determination of the relative prediction error caused by uncertainty in the data used realized manufacturing times of the products. Since this information is only available after the product has been manufactured, this approach is unsuitable for the prediction of new products. It is expected that the prediction error for new products will be higher than 0.25. Therefore, the objective for this research is set to reduce the **MAPE** for predicting manufacturing times of products from 0.60 to 0.30.

1.4. Research question

Based on the above mentioned research objective, the following research question can be formulated:

"How to develop a manufacturing time prediction model, using ***BIM*** *and manufacturing data, in order to create more effective manufacturing schedules?"*

Before this research question can be answered, several sub-questions are to be answered:

- What is the current state at Oostingh Staalbouw?
- Which data can be extracted from **BIM** and the manufacturing process?
- Which manufacturing time prediction models are available in literature?
- Which conceptual prediction model can best be used to predict manufacturing times?
- How can the performance of prediction models be evaluated and compared?
- How can the conceptual prediction model be validated?

1.5. Research scope

Due to both limited available data and time constraints, this research is subject to several boundaries.

- **Given condition**: The case study is conducted at Oostingh Staalbouw. Oostingh Staalbouw is specialized in the steel structural part of construction projects. Available data for this case study are physical properties stored in **BIM** and manufacturing times of products.
- **Assumptions**: It is assumed that Oostingh Staalbouw is representative for the manufacturing of (steel) structural elements. Furthermore, constrained by available data, it is assumed that manufacturing times are not influenced by process related and external factors like weather, day of the week, etc.
- **Limitations**: The validation of this research is limited to a case study for the manufacturing of steel elements. Therefore the generality of the prediction model can only be validated in a hypothetical situation. In addition, the validation of the proposed prediction model is limited to the assembly and welding step. At last, this research is limited to proposing, modelling and validation of the manufacturing time prediction model. The formulation of manufacturing schedules using the predicted manufacturing times is left out of scope in this research.

1.6. Methodology

The methodology used in this research is based on the Standard Process for Data Mining (CRISP-DM) [10]. According to these guidelines, a predictive data analytical project lifecycle consists of six key phases; business understanding, data understanding, data preparation, modeling, evaluation and deployment.

This methodology remains ambiguous on how to determine beforehand which prediction model would best apply to the problem. Therefore, in this research an update to this methodology is proposed and implemented. For this research, reviewing literature for approaches in related problems is added to the methodology. Additionally, emphasis is put on evaluating prediction models based on theory. Based on this evaluation, the prediction model most likely to suit the defined problem is derived. This approach is opposed to the trial and error approach currently used in the industrial standard for the selection of a prediction models.

First, the situation for the case study at the company will be explored; The manufacturing process will be mapped and relevant aspects of the manufacturing process will be highlighted. Afterwards, the characteristics of the available data will be investigated through data analysis. Based on identified characteristics of the data, prerequisites for the prediction model will be set. Then, a literature study will be conducted on integrating **BIM** in manufacturing processes. Furthermore, approaches for predicting manufacturing times in other **ETO** industries will be studied. Based on the characteristics identified for the case study, along with the studied literature, a conceptual prediction model will be proposed. Afterwards, this conceptual model will be validated in both hypothetical and real case scenarios.

1.7. Structure of the report

At first, chapter 2 provides insight on the current state of the company for the case study, Oostingh Staalbouw. Chapter 3 reviews literature on predicting manufacturing times. Based on both available data and prediction models, chapter 4 proposes a general prediction model for predicting manufacturing times using **BIM**. The proposed prediction model is verified and validated in chapter 5 and at last, the conclusions and recommendations of this research are presented in chapter 6.

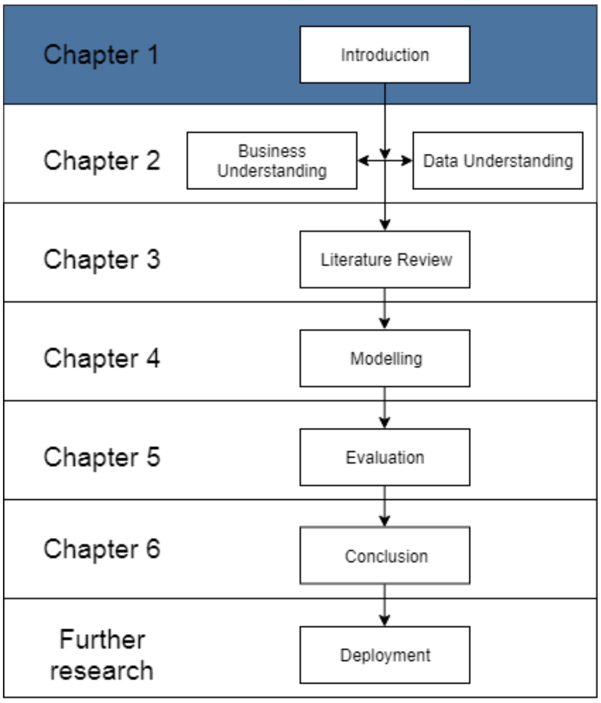


Figure 1.5: Overview of methodology used in this research, along with corresponding chapters in this report

2

Current state at Oostingh Staalbouw

2.1. Introduction

In order to provide the reader an understanding of the background for this research, this chapter will discuss key aspects of Oostingh Staalbouw, the company studied during this case study.

The major competitive advantage of Oostingh Staalbouw is that all aspects of the construction process for steel structures are handled. From the design phase to manufacturing and the erection of the structure at the building site are carried out by the company.

This research focuses on the relation between the design phase (physical properties stored in **BIM**) and the manufacturing process at Oostingh Staalbouw (manufacturing times). This relation will be used to predict manufacturing times based on the design (physical properties) of unique products.

During the design phase of a project, a Building Information Model (**BIM**) is used by Oostingh Staalbouw. **BIM** is used to store information about structural elements of a construction project as well as information about the manufacturing process. The first part of this chapter will elaborate on what **BIM** is, and which data is currently stored in these models.

Afterwards, an overview of the manufacturing process at Oostingh Staalbouw is provided. Onwards, the collected data from the manufacturing process will be discussed.

This way both the subquestion *"What is the current state at Oostingh Staalbouw"* and *"Which data can be extracted from **BIM** and the manufacturing process"* are aimed to be answered.

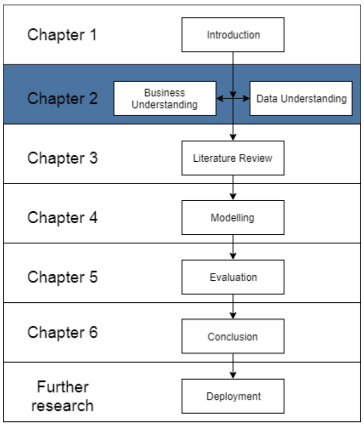


Figure 2.1: Second and third step of the used methodology for this research

2.2. Building Information Modelling

In order to provide the reader a better understanding of Building Information Models, this section will discuss **BIM** in more detail. At first, a definition of **BIM** is provided, along with relevant literature on **BIM**. Afterwards, the information stored in **BIM**, available for this case study is discussed.

2.2.1. Definition of BIM

Building Information Modelling (**BIM**) technology can be used to digitally create an accurate virtual model of a construction project. Upon completion, the model incorporates the precise geometry and other relevant information, like construction order of the construction [18]. The model can be used for cooperation of all involved members of the construction project during the entire lifetime of a building. Relevant information from designing, manufacturing and assembling up to maintenance of the building can be stored in the model. According to Eastman et al. [18], **BIM** can be described as:

> *"With BIM (Building Information Modelling) technology, one or more accurate virtual models of a building are constructed digitally. They support design through its phases, allowing better analysis and control than manual processes. When completed, these computer-generated models contain precise geometry and data needed to support the construction, fabrication, and procurement activities through which the building is realized."*

Basically, **BIM** can be used to store information about each stage of the life cycle of a construction project. All sorts of information can be coupled to either a single structural element or (a part of) the entire building and the information can be accessed by all collaborators of the project. A visualization of **BIM** and it being the spine of construction projects is shown in Figure 2.2

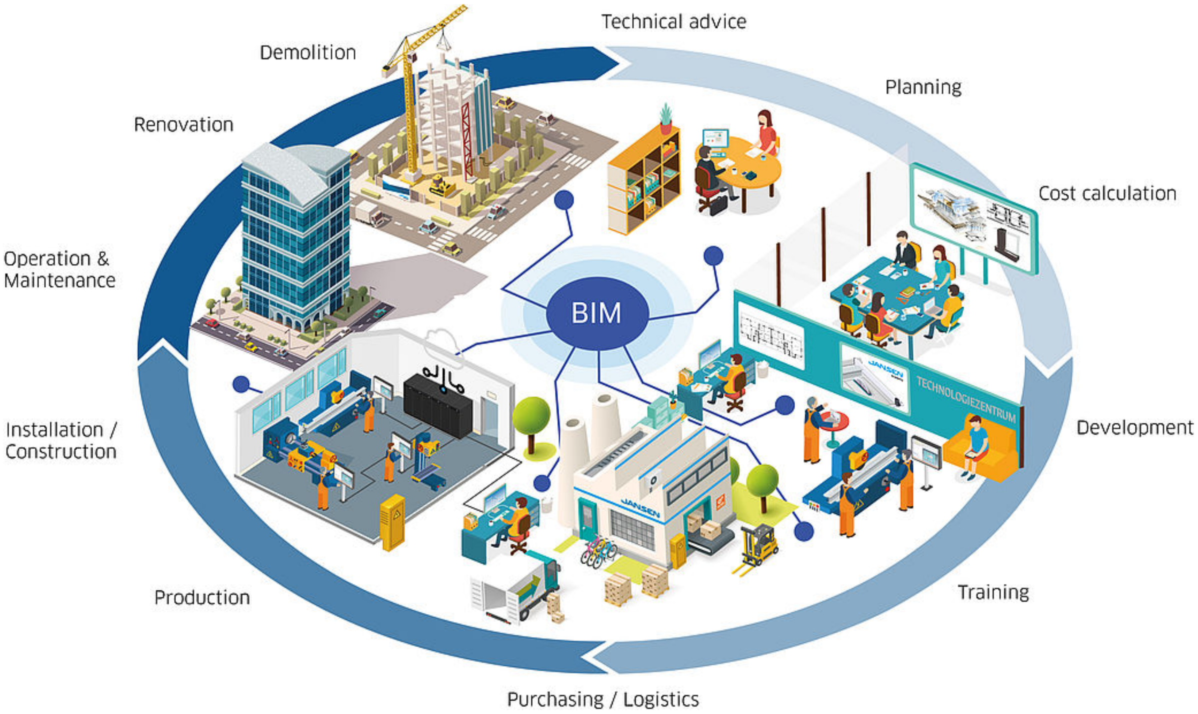


Figure 2.2: Visualization of the interconnectivity of BIM in the life cycle of a building [30]

In theory, all kinds of information can be stored in **BIM**. However, the use of **BIM** is still mainly used in the design phase of construction projects, limiting the stored information to physical properties of the project [26]. Oostingh Staalbouw is no exception to this rule.

2.2.2. Related literature on using BIM for scheduling activities

Prior research on the implementation of **BIM** for scheduling activities focused on integrating the planning between manufacturing and assembling [55] [44], and is extended to the integration of the design phase of construction projects in planning activities [47]. Here the manufacturing process has been considered a black box and ignores detailed manufacturing schedules.

Implementing information from **BIM** in on-site assembly planning of prefabricated components have been addressed extensively by **?** [**?**]. The manufacturing of the prefabricated components, however, was left out of scope in their research.

Predicting required man-hours for assembly on the construction site has been investigated by Lee et al. [36]. In this research, the construction hours are predicted using a bill of materials, which is extracted from **BIM**. Comparing different trendlines, they succeeded in making reliable predictions for required assembly times.

Significantly less research has been conducted on using information from **BIM** for scheduling of off-site manufacturing processes. Liu et al. [39] created a discrete event simulation for an off-site panel construction factory in order to assist manufacturing planning. The physical aspects of the panels are extracted from **BIM** and used as input for the simulation. During this research, however, the manufacturing times of the panels were not predicted based on information from **BIM**. The manufacturing times of the several workstations were estimated based on experience and kept deterministic, leading to a less realistic simulation.

An approach for man-hour estimation based on data from **BIM** and the manufacturing process has been proposed by Hu et al. [26]. In this approach, physical properties of structural elements were extracted from **BIM**. Using data from a real manufacturing process, evaluation of these different properties with respect to the required manufacturing time was performed. In this case study, a Multiple Linear Regression (**MLR**) approach increased the accuracy of predicting required manufacturing times, compared to a traditional, experience based approach.

Mohsenijam and Lu [40] used the prediction model from Hu et al. [26] during the design phase of building projects. Through linking man-hour estimations with the design process, the study gained more insight in the consequences of design choices for the required manufacturing times.

2.2.3. Data stored in BIM

At the start of each construction project, limited information is available. Before a project commences, an initial drawing of the design of the structural construction is prepared. This fundamental drawing contains information about the type. length and placement of the steel profiles that will be used.

As the design phase of the project progresses, details of the design are filled in, and more information about the physical properties of the products becomes available. This information is stored in the Building Information Model (**BIM**). Information can be extracted from this **BIM** either per product or per (part of a) project. An example of extracted data on product level from **BIM** is provided in table 2.1.

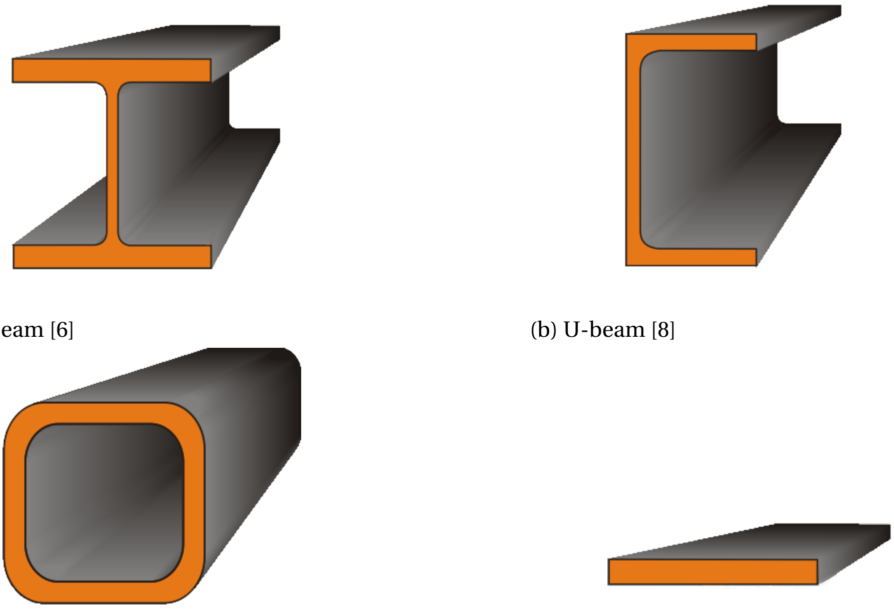
Table 2.1: Example of extracted data from **BIM**

Product	Count	Main profile	Weight product [MT]	Length product [mm]	Part number	Profile part	Length [mm]	Weld thickness	Hole diameter [mm]	Amount
L570	2						33		22	1
L570	2						90		33	4
L570	1						233	5		1
L570	2						103	5		1
L570	2						300	5		1
L570	2						315	6		1
L570	4						300	6		1
L570	1	HEA360	0,394	3.158	L570	HEA360	3.118			1
L570	1	HEA360	0,394	3.158	P381	ST15X300	330			1
L570	2	HEA360	0,394	3.158	P42	ST20X300	400			1
L570	1	HEA360	0,394	3.158	P543	PL10X143	313			1

The extracted data can be divided in both categorical and quantitative variables. Products consist of a main profile, based on standard profile types. Four categories of standard profiles can be distinguished:

1. H-beam (Figure 2.3a)
2. U-Beam (Figure 2.3b)
3. Hollow section (Figure 2.3c)
4. Plate (Figure 2.3d)

It is expected that the handling of these different profiles varies slightly. In order to investigate the effect of the different profile types on manufacturing time, the difference in profile type will be taken into account by the prediction model. This leads to the prerequisite that the prediction model must be able to cope with categorical input variables.



(a) H-beam [6]

(b) U-beam [8]

(c) Hollow section [7]

(d) Plate [6]

Figure 2.3: Visualization of the identified profile types

In addition, the following quantitative physical properties extracted from **BIM** can be identified:

- Total weight of the product
- Maximum length of the product
- Total weld length
- Number of parts
- Number of welds
- Number of holes

Data preparation
The extracted data is unprocessed data and should be prepared before it can be used properly. As shown in table 2.1, each product consists of one or multiple welds, parts, holes etc. This data should be summarized for each product. The weight and length of the products can be extracted directly. The number of parts, holes and welds should be summed up per product.

The total weld length is calculated by taking into account the thickness of the welds. Different weld thicknesses require different number of layers to weld. It is assumed that each layer requires equal welding time. Table 2.2 shows the relationship between weld thickness and the number of weld layers.

Table 2.2: Table of weld thickness and corresponding number of weld layers

Weld thickness	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	30
No. weld layers	1	1	1	3	3	4	6	6	6	10	10	15	15	21	21	27	27	34	51

2.3. Manufacturing process

After the design phase of a construction project, the designed products are manufactured. In this section, the outline of the manufacturing process at Oostingh Staalbouw is discussed. Afterwards, the collection of data from the manufacturing process is considered.

2.3.1. Outline of the manufacturing process

The manufacturing process at Oostingh Staalbouw can be divided in preprocessing, assembly, welding and coating of products. Figure 2.4 shows a flow diagram of the manufacturing process at Oostingh Staalbouw. Noticeably, there is a storage step between each manufacturing step. These storage areas function as buffers in order to compensate for disrupted flow of products due to ineffective manufacturing schedules. The physical properties of the products (on average a product weights 400 kg and is 4000 mm long) lead to significant transfer and spatial costs for the use of these buffers.

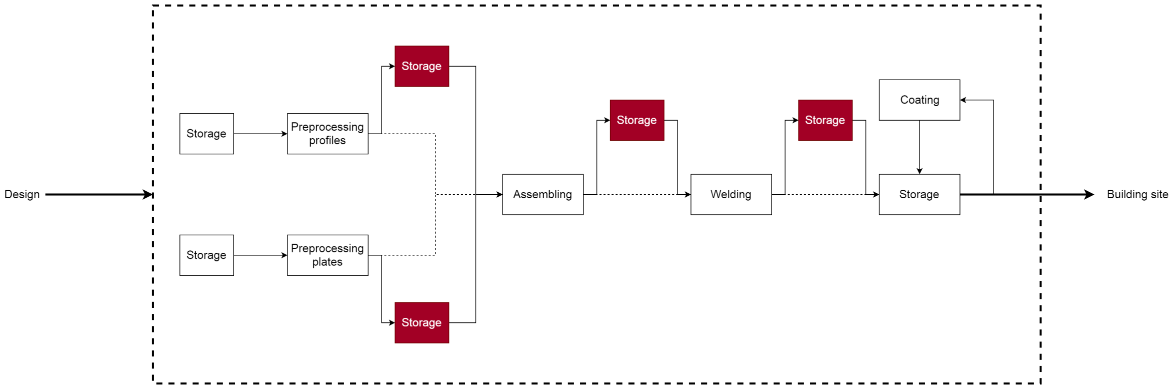


Figure 2.4: Manufacturing flow at Oostingh Staalbouw. Notable are the storages (red boxes) between the subsequent manufacturing steps.

Due to the Engineered-to-Order (**ETO**) nature of the construction projects carried out at Oostingh Staalbouw, in combination with the complexity of manufacturing operations makes it a process difficult to automate. Therefore humans are highly involved in the assembling, welding and coating phase. Inherent to this extensive human involvement, uncertainty in manufacturing times is expected to arise.

In the remainder of this section, the various steps of the manufacturing process will be discussed in detail.

Preprocessing

Standard parts are supplied from steel producers. An example of standard parts are H-beams with a standard length (6 or 12 meters). A major distinction can be made between steel profiles (H-beam, U-beams and Hollow Sections) and steel plates. During preprocessing, these standard parts are processed into custom sizes, according to the requirements set in the design phase of the construction project.

Steel profiles

In the preprocessing stage of steel profiles, standard steel profiles are cut to the required length. Two different Computer Numerical Control (**CNC**) machines are used parallel to each other; a torch cutting machine and a sawmill. Each machine is capable of cutting a steel profile to the required length and drilling holes in the profile. After the profile is processed, it is placed in a storage area. This storage area is divided into sections, in which each section represents a truckload. A flow diagram of this manufacturing step is provided in Figure 2.5

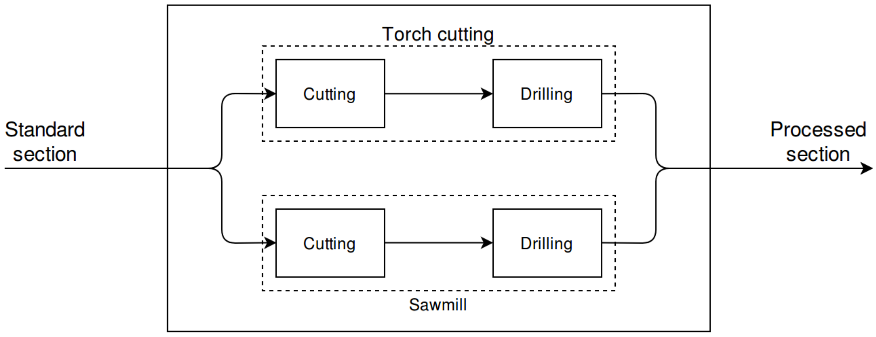


Figure 2.5: Flow diagram of the preprocessing of steel profiles at Oostingh Staalbouw

Steel plates

Similar to the preprocessing of steel profiles, custom steel plates are cut from standard steel plates by **CNC** machines. These **CNC** machines use a laser to cut the steel plates in any arbitrary shape.

Assembly

During the assembly step, the preprocessed steel profiles and plates are joined according to the working drawing. Each workplace consists of a workbench and a frame to put the product on. The workbenches are standardized using the 5S principle [37] and can be used for all assembly operations. The parts to be assembled are placed next to the assembly workbench (Figure 2.6). Each assembly employee is competent for all assembly operations. Products, however, vary in length from 0.2 up to 20 meters. Therefore, one product can occupy multiple workplaces.

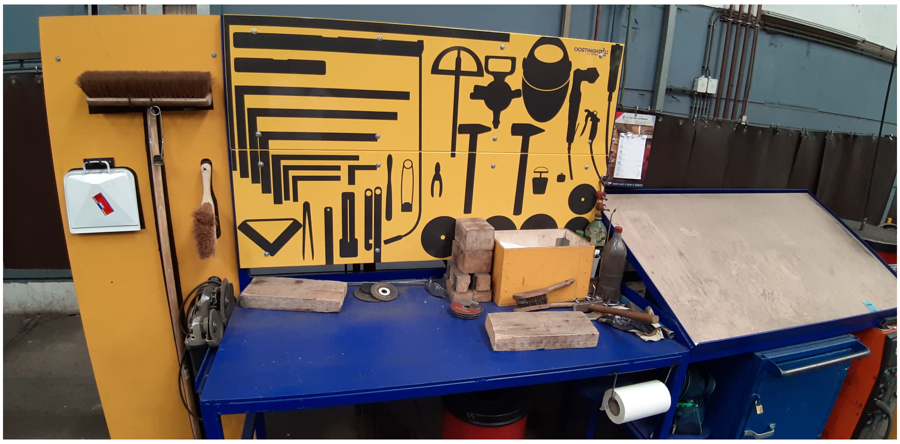


Figure 2.6: Example of a standardized assembly workbench at Oostingh Staalbouw

The next steel profile to be assembled is transported to an empty assembly workplace by an overhead crane. The corresponding steel plates are transported to the particular workplace either by hand or forklift truck.

Preprocessed parts of a product are transported to an empty assembly workplace. An assembly employee is assigned to this product and assembles the product according to the working drawing, specified in the design phase. The assembly step is highly dependent on both the preprocessing and design phase of the projects; if a defect occurs in one of the custom parts, this has to be corrected during the assembly step. Additionally, in case of unclear working drawings resuling from errors in the design phase, investigation is required during the assembly step. Therefore, it is expected that the manufacturing step is most sensitive to human related uncertainty in manufacturing times.

After the product is assembled, an overhead crane is used to transport the product to a storage area between the workplaces, before it is transported to the workplace where the next manufacturing step will be executed.

The flow of products in the assembly step is shown in Figure 2.7.

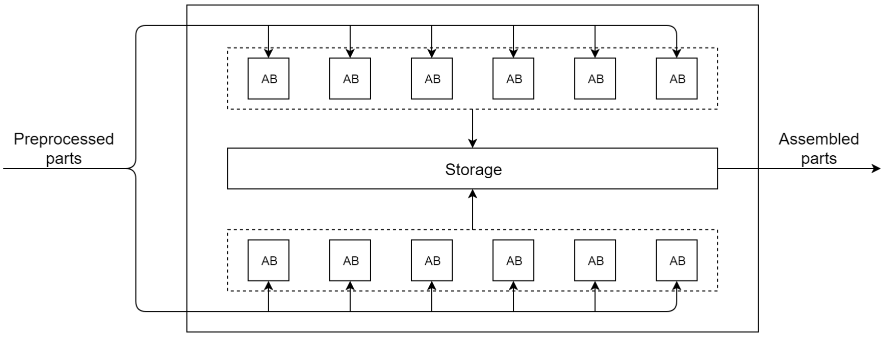


Figure 2.7: Flowdiagram of the assembly step in the manufacturing process at Oostingh Staalbouw

Welding

After the separate, custom parts are assembled into one product, the product is transported from storage to the welding step. The layout of this welding step is divided into two columns, each existing of multiple welding workplaces. Analogous to the assembly step, each workplace is standardized. Therefore each product can be welded at each workplace, it should however, be taken into account that the number of occupied workplaces is dependent on the length of the product.

During the welding step, the parts connected through a spot weld in the assembly step are welded with final welds. The prescribed quality and size of welds vary per product, but can vary for several parts of one product as well.

After the product has been welded it is transported to a storage area.

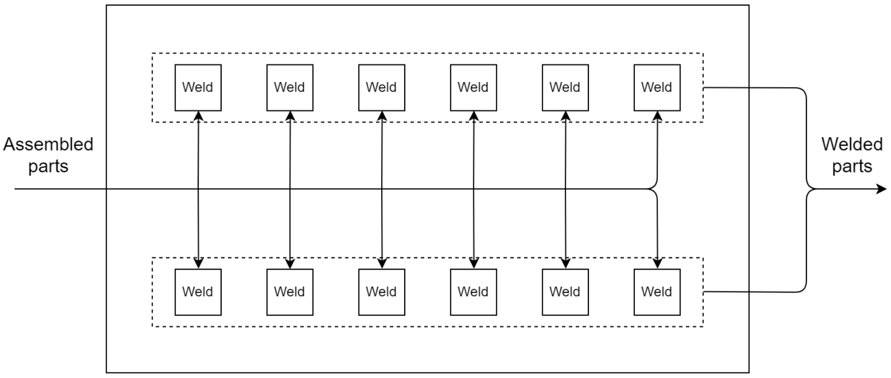


Figure 2.8: Flowdiagram of the welding step in the manufacturing process at Oostingh Staalbouw

Coating

In order to prevent oxidation of the products, most products are coated. This process exists of two steps; the blasting and painting of the products. The product is placed in a blasting machine, in which the surface of the products is roughened. After the surface is prepared, the product is placed in a painting room, in which a coating is applied. Afterwards, the product is left in the room until the coating has dried. A schematic overview of this process is provided in Figure 2.9.

Due to the process characteristics of the coating step at Oostingh Staalbouw, the manufacturing time for this step is not registered on product base. Therefore, the coating step is left out of scope for this research.

Focus of this research

Currently, the assembly and welding step account for 70% of manufacturing costs at Oostingh Staalbouw. Furthermore, Oostingh Staalbouw currently has most difficulties predicting these manufacturing steps accurately. Therefore, for the remainder of this research, emphasis will be put on the prediction of manufacturing times for the assembly and welding step.

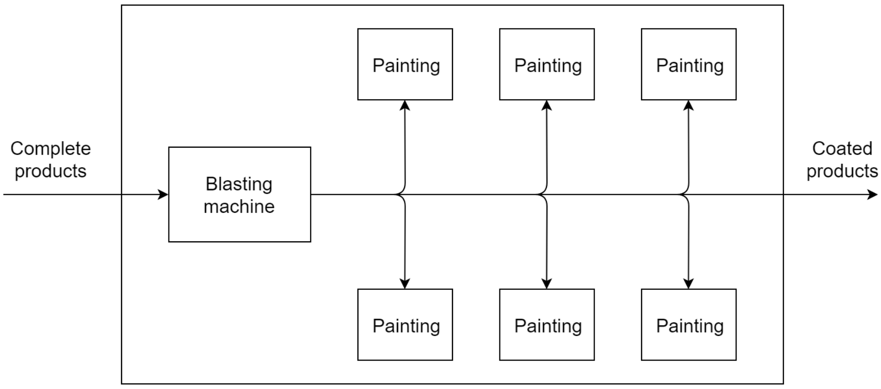


Figure 2.9: Flowdiagram of the coating process at Oostingh Staalbouw

2.3.2. Data collection for the manufacturing process

Data from the manufacturing is collected in two ways. The preprocessing steps uses CNC machines. These machines automatically store relevant information, like manufacturing time.

Information labels with barcodes are used for the manual manufacturing steps. On these labels, information is written about the weight of the product, the required weld quality, applied coating, which project it belongs to and on which truck it should be loaded. An example of an information label is provided in Figure 2.10

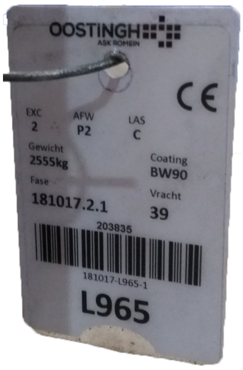


Figure 2.10: Example of information label used to measure product manufacturing times per manufacturing step at Oostingh Staalbouw

These labels are scanned at the beginning and at the end of the manufacturing step. The manufacturing time and employee ID are subsequently coupled to each product. This data is synchronized with **BIM** after each work shift.

An example of unprocessed, exported data is provided in Table 2.3. In case multiple workshifts are required to finish the manufacturing step for a product, the registered hours are summed up.

Table 2.3: Example of data extracted from the manufacturing process

Employee ID	JobDescription	Hours	Product ID
466	Welding	0,23	181017-L502-1
667	Welding	1,2	181017-L202-3
406	Welding	2,15	184030-VWS2-1
823	Welding	2,18	184030-VWS2-1
463	Welding	2,55	181017-L178-1

2.4. Data analysis

Collected data from both design and manufacturing can be analysed in order to look for relationships in this data. In this section the data analysis on the collected data from both design and manufacturing stage is considered.

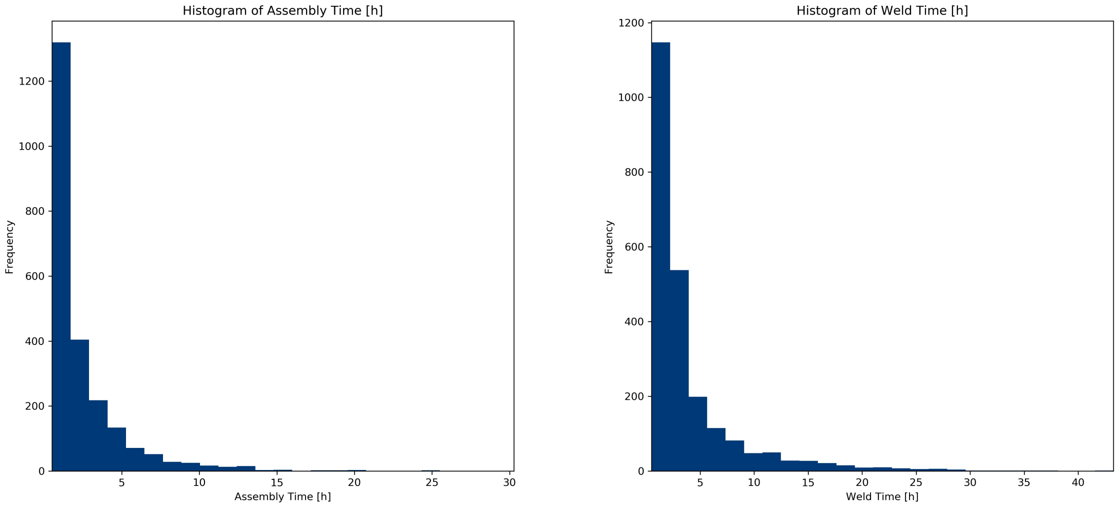
2.4.1. Correlation analysis

In order to test whether a relationship exists between quantitative physical properties and the manufacturing time, a correlation coefficient is used. Two main types of correlation coefficients are commonly used; the Pearson's correlation coefficient and the Spearman's correlation coefficient. Pearson's correlation coefficient is favorable under the assumption that the data is normally distributed. The histograms of the assembly times (Figure 2.11a) and welding times (Figure 2.11b) show that the data is not normally distributed, therefore Spearman's correlation coefficient should be used. Using Spearman's correlation coefficient, for each variable the data is ranked. To determine the correlation between two variables, the difference between the ranks d_i is summed over all data points (n) and put in equation 2.1. Values close to 1 indicate strong monotonic positive correlation between two variables. [41]

$$r_s = 1 - \frac{6\sum_{i=1}^{n} d_i^2}{n(n^2-1)} \tag{2.1}$$

The correlations between the physical properties and the manufacturing times at the assembly and welding step are provided in table 2.4. The relatively high correlation coefficients between the various physical properties and the manufacturing times hints at a relationship between these variables.

The major drawback of using Spearman's rank coefficient is that only shows that a monotonic relationship exists between two variables; if one variable increases, the other variable will increase as well. The relationship, however, can be either linear or nonlinear [41].



(a) (b)

Figure 2.11: Histogram of realised a) assembly and b) welding times at Oostingh Staalbouw. The non normal distribution of manufacturing times violates the assumption required for Pearson's correlation analysis.

Table 2.4: Derived Spearman correlation coefficient between physical properties and manufacturing times

	Weight	Length	Weld length	No. Parts	No. Welds	No. Holes	Assembly Time	Weld Time
Weight	1.0	0.87	0.93	0.82	0.73	0.61	0.80	0.88
Length	0.88	1.0	0.80	0.71	0.58	0.53	0.73	0.74
WeldLength	0.94	0.80	1.0	0.84	0.77	0.66	0.80	0.92
No. Parts	0.82	0.71	0.84	1.0	0.86	0.72	0.83	0.86
No. Welds	0.74	0.59	0.77	0.86	1.0	0.64	0.82	0.81
No. Holes	0.62	0.54	0.66	0.72	0.64	1.0	0.64	0.68
Assembly Time	0.80	0.74	0.80	0.83	0.82	0.64	1.0	0.81
Weld Time	0.89	0.75	0.92	0.86	0.81	0.68	0.81	1.0

2.4.2. Uncertainty in the data

As stated in section 2.3, human operators are highly involved in the manufacturing process. Inherent to human involvement, uncertainty in manufacturing times is expected to be encountered. An example of this uncertainty in manufacturing time is provided in table 2.5, along with the corresponding graph in Figure 2.12. The data is this example relates to a product manufactured multiple times.

Table 2.5: Example of fluctuations in manufacturing times for an identical product, manufactured multiple times

	1	2	3	4	5	6	7	8	9	10	11
Assembly Time [h]	0,57	1,67	1,63	1,03	0,97	0,67	1,43	0,63	0,55	1,63	0,48
Weld Time [h]	1,15	1,89	0,85	2,06	1,98	1,55	1,67	1,42	2,03	1,01	0,82

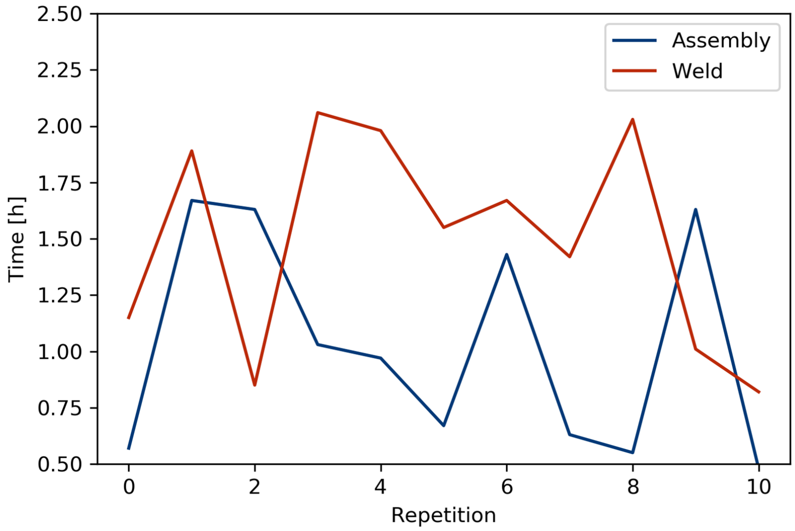


Figure 2.12: Example of uncertainty in the data for an identical product manufactured multiple times

Due to these fluctuations, a prediction error will always arise: the physical properties of these products are identical, ergo the predicted manufacturing time will be identical as well. Since the realised manufacturing times fluctuates, a deviation from the predicted manufacturing time will occur. Gaining insight in the size of this error, the available data of products that are manufactured multiple times is evaluated. For each product manufactured multiple times, the manufacturing time is predicted using the mean manufacturing time of the repetitions of this product. The Mean Absolute Percentage Error (**MAPE**), Median Absolute Percentage Error (**MeAPE**) and the Standard Deviation of the Absolute Percentage Error (**Std**) (Section 5.2.1) for both the assembly and welding manufacturing step are summarized in table 2.6.

The results in table 2.6 imply that the possible prediction accuracy using this approach is 0.25 (25%). Since this approach requires historical information of manufacturing times of the product, it is unsuitable to predict manufacturing times for new, unique products. It is expected that prediction models will yield a **MAPE**

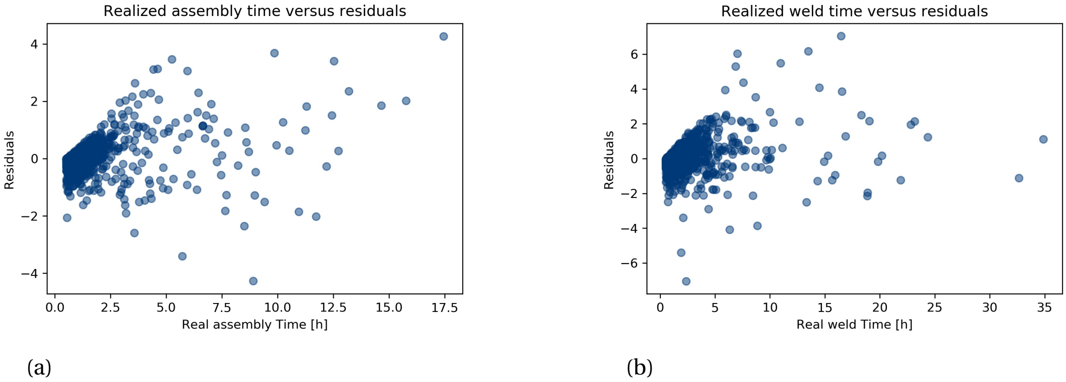
Table 2.6: Summary of absolute relative errors in data due to human related factors

Mean	Median	Std	C_v
0.25	0.19	0.24	0.96

higher than 0.25. Therefore, the objective **MAPE** of this research is set to 0.30. This would be a significant improvement compared to the **MAPE** of the currently used approach (0.60). Additionally, the Standard Deviation of 0.24 gives insight in the distribution of the prediction errors for identical products.

The Coefficient of Variance ($C_v = \frac{Std}{Mean}$) close to one implies that the relative uncertainty is relatively high and follows an exponential distribution [1], [15]. The exponential distribution of absolute relative errors implies that (significant) outliers are present. This leads to the significant difference between the Mean and Median absolute relative error.

Figure 2.13 shows the residuals of the above mentioned approach versus the realized manufacturing times. From these plots, it can be seen that the variance of the residuals is not constant for different manufacturing times. This phenomenon is called heteroskedasticity [21]. In this case the variance of the residuals increase with increasing manufacturing times. This can be explained by external variables having a relative influence on the manufacturing times of a product. For example, due to experience an employee might be able to weld 10 % faster than inexperienced colleagues, this effect will amplify with increasing manufacturing times.



(a) (b)

Figure 2.13: Residual plot versus realized manufacturing time for a) assembly and b) welding for the estimation of uncertainty in manufacturing times.

2.4.3. Prerequisites for the prediction model

Based on the data analysis, several data characteristics can be identified. These data characteristics are used to set prerequisites for the selection of a proper prediction model for the prediction of manufacturing times per manufacturing step.

- **Quantitative and categorical input variables**; two types of physical variables are identified, quantitative variables (length, weight, etc.) and the categorical variable profile type. Therefore the prediction model should be able to cope with both quantitative and categorical input variables.
- **Linear and nonlinear relationships**; the correlation analysis between physical properties and manufacturing times remained ambiguous whether the relationship between these variables is linear or nonlinear. Additionally, considering the desired generality of the manufacturing time prediction model, the proposed prediction model should be able to yield accurate predictions for both linear and nonlinear relationships.
- **Uncertainty in the data;** due to human involvement, significant uncertainty in the data is present. The prediction model should be robust against this uncertainty.

2.5. Predictions of manufacturing times at Oostingh Staalbouw

Currently, manufacturing scheduling at Oostingh Staalbouw uses experience based estimations of manufacturing times by manufacturing managers. These estimations are produced for a group of products, which is common for the construction industry [26].

In the recent past (2017), Oostingh Staalbouw conducted an experiment where manufacturing times were predicted per product, rather than per group of products. This experiment resulted in a Mean Absolute Percentage Error (**MAPE**) of 0.60 for the experience based predictions of manufacturing times per product (over a period of a year). After this experiment, Oostingh Staalbouw returned to the original approach of predicting manufacturing times per groups of products. It is, however, assumed that the found **MAPE** is still representative for the prediction approach used in the manufacturing process.

2.6. Conclusion

In this chapter, the reader is provided an understanding of the current state at Oostingh Staalbouw, the company used as case study for this research. Both the subquestion *"What is the current state at Oostingh Staalbouw"* and *"Which data can be extracted from **BIM** and the manufacturing process"* are discussed in this chapter.

The relationship between the design phase and the manufacturing process is the main focus area for this research. During the design phase, a Building Information Model (**BIM**) is used to create a digital copy of the construction. From this model, physical properties can be extracted. These properties can be divided in both quantitative and categorical variables.

The manufacturing process consists of a preprocessing, assembly, welding and coating step. Apart from the preprocessing step, the manufacturing steps consists mainly of human labour, leading to uncertainty in manufacturing times. For the assembly and welding step, the manufacturing time per manufacturing step is collected by scanning barcodes coupled to the product. No detailed information of manufacturing times per product in the coating step is available. Therefore, in the remainder of this research, the focus will be on the assembly and welding step at Oostingh Staalbouw.

Using Spearman's correlation rank, a monotonic relationship between quantitative physical properties and the manufacturing time per manufacturing step can be identified. The major drawback of this correlation analysis is that it remains ambiguous whether the relationships are linear or nonlinear.

Based on a data analysis prerequisites for the prediction model can be derived. The prediction model should be able to incorporate both quantitative and categorical input variables. Additionally, the prediction model should be able to yield accurate results for both linear and nonlinear relationships. At last, uncertainty in the data is identified. Therefore, the prediction model should be robust against this uncertainty.

3

Literature Review

3.1. Introduction

Before a conceptual model for predicting manufacturing times based on data from **BIM** and the manufacturing processes can be developed, a literature review is conducted. This chapter aims at answering the sub-question: *"Which manufacturing time prediction models are available in literature?"*.

Before relevant literature is to be reviewed, it should be noted that the prediction problem identified for this research is called a regression problem. Regression models aim to predict a continuous variable, using the relation between several different input variables [48]. In the research area of regression, several different prediction models can be identified.

In order to get an idea of implemented regression models in related research, a literature review is conducted. Afterwards, the identified prediction models are explained in more detail. Based on the prerequisites set for the prediction model (chapter 2) and the reviewed literature, a conceptual prediction model will be proposed.

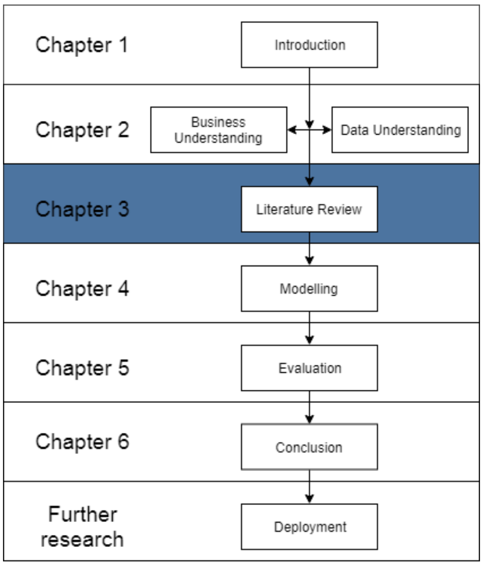


Figure 3.1: Fourth step of the used methodology for this research

3.2. Manufacturing time prediction in related industries

As stated in section 2.2.2, prediction of manufacturing times based on historical data from **BIM** and the manufacturing process is limited. Therefore, approaches in other industries with Engineered-to-order (**ETO**) characteristics are reviewed. In this section, the related literature is discussed.

Tirkel [54] implemented a Neural Network (**NN**) model to predict the product lead time of wafer fabrication. Historical information like the week of manufacturing, product name and the required tools are fed into the model, along with the realised lead times. The accuracy of the prediction model turned out to be highly dependent on lead time duration, with accurate results on relatively short lead times and increasing error with increasing lead time.

Pfeiffer et al. [45] integrated Multiple Linear Regression (**MLR**) and Tree Based Regression (**TBR**) prediction models with a discrete event simulation of a manufacturing process in order to predict product lead times. Based on the results of the discrete event simulation, the prediction models are trained and evaluated. In this research, both **MLR** and **TBR** turned out to be capable of predicting product lead time accurately. Application in a real case scenario, however, was not considered during this research.

Hur et al. [27] compared **MLR** with a **TBR** approach on a quarterly, monthly and daily estimation basis for predicting the lead time of a shipbuilding process. As expected, both models became more accurate as the timespan became shorter. Furthermore, the **TBR** approach slightly outperformed the **MLR** approach.

The combination of a Support Vector Regression (**SVR**) with a particle swarm optimization algorithm in order to find the best parameters for the **SVR** algorithm is investigated by Yu and Cai [58]. In this study, the final model was able to predict required man-hours for aircraft assembly with high accuracy.

Selecting the right input variables for **MLR** through stepwise regression is discussed in Arash et al. [4]. In this research, two case examples of the implementation of **MLR** are provided; the first case study aims at predicting the amount of slump of a concrete mix. The second case is focused on predicting one-span installation cycle-time of a precast viaduct construction. In both cases, the required number of inputs could be reduced by stepwise regression, while keeping an accurate prediction model.

MLR, **TBR** and **SVR** have been compared for use by an optical glass manufacturer by Gyulai et al. [23]. In this study, **TBR** and **SVR** slightly outperformed the **MLR** approach in terms of accurately predicting product lead times.

Nagahara and Nonaka [43] compared standard **MLR** with Ridge Regression and experience based predictions for product lead time of a variety of semiconductors. The prediction is based on product-specific values, model parameters and measured manufacturing times. Both standard **MLR** and the Ridge Regression approach outperformed the experience based predictions significantly.

Common prediction models are compared by Lingitz et al. [38] for predicting product lead time in a semiconductor manufacturing process. The Random Forest approach, which is an adaption of the **TBR** approach turned out to be slightly more accurate than the other approaches in this research.

Lastly, as stated in section 2.2.2, Hu et al. [26] and Mohsenijam and Lu [40] use **MLR** for prediction of required man-hours of products based on data from **BIM** and the manufacturing process.

The conducted literature research on prediction models for making manufacturing process related predictions is summarized in table 3.1.

- MLR: Multiple Linear Regression
- TBR: Tree Based Regression
- SVR: Support Vector Regression
- NN: Neural Network

Table 3.1: Overview of implemented prediction models in reviewed literature

Reference	Year	Industry	MLR	TBR	SVR	NN
Tirkel [54]	2013	Semiconductor				X
Hu et al. [26]	2014	Prefabrication	X			
Pfeiffer et al. [45]	2015		X	X		
Hur et al. [27]	2015	Shipbuilding	X	X		
Yu and Cai [58]	2015	Aircraft				X
Mohsenijam and Lu [40]	2016	Prefabrication	X			
Arash et al. [4]	2017	Construction	X			
Lingitz et al. [38]	2018	Semiconductor	X	X	X	X
Nagahara and Nonaka [43]	2018	Semiconductor	X			
Gyulai et al. [23]	2018	Optical	X	X	X	

Table 3.1, shows that **MLR**, **TBR**, **SVR** and **NN** are common choices for predicting manufacturing times. In the next section of this chapter, the underlying theory behind these prediction models will be discussed in more detail. Due to time limitations, this research is limited to the basic prediction models and will not take into account all varieties of the discussed models.

It should be noted that the covered literature focuses on predicting product lead times, rather than manufacturing times per manufacturing step. The difference between the product lead time and the manufacturing time per manufacturing step is that the product lead time is defined as *"the time required once the product began its manufacture until the time it is completely processed"* [9]. The manufacturing time per manufacturing step is the time it takes for the product to complete one step of the entire manufacturing process. Therefore, the latter is especially useful for the formulation of manufacturing schedules. To the knowledge of the author, literature focusing on the prediction of manufacturing time per manufacturing step is limited to the research by Hu et al. [26].

Relevant input variables

Based on the covered literature, relevant input variables can be identified. Overall, a distinction can be made between product related variables, process related variables and external influence variables.

Across the covered literature, studies focused on product related (phyiscal) variables for the prediction of manufacturing time, while process related variables (occasionally in combination with external variables) are used for the prediction of product lead times.

For the prediction of product lead time, manufacturing times of the various manufacturing steps are approximated using the mean manufacturing time for a category of products. This approach, however, is limited to manufacturing processes with small variations between different products.

Figure 3.2 shows the identified categories of input variables. Each category contains examples of related input variables. Note that the possible variables are depending on the characteristics of the manufacturing process. Therefore, the possible variables are not limited to the examples shown in this Figure.

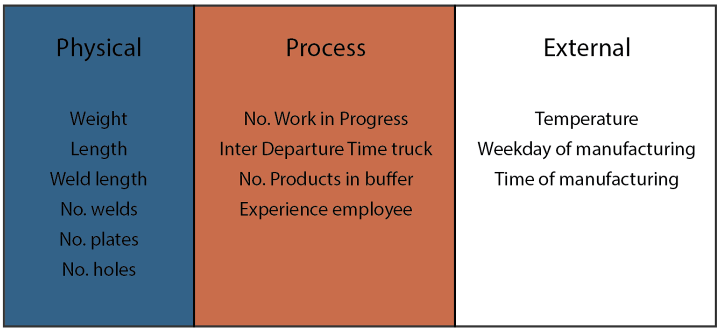


Figure 3.2: Identified categories of input variables, each category contains examples of related input variables

As discussed in chapter 2, available data for this case study is limited to physical variables. Based on prior research by Hu et al. [26], it is assumed that the absence of both process and external variables will have limited influence on the accuracy of predicting manufacturing times.

3.3. Comparison of prediction models

In this section, the prediction models pointed out in Section 3.2 will be discussed in more detail. It should be noted, that due to time constraints, this research is limited to the standard prediction models, rather than reviewing all adaptions proposed throughout literature. Afterwards, the different prediction models are compared using the requirements derived in section 2.4.3.

- **Linear and nonlinear relationships**: the correlation analysis between physical properties and manufacturing times remained ambiguous whether the relationship between these variables is linear or nonlinear. Additionally, considering the desired generality of the manufacturing time prediction model, the proposed prediction model should be able to yield accurate predictions for both linear and nonlinear relationships.
- **Uncertainty in the data**: due to high human involvement, significant uncertainty in the data is present. The prediction model should be robust against these fluctuations.
- **Quantitative and categorical input variables**: two types of physical variables are identified, quantitative variables (length, weight, etc.) and the categorical variable; profile type. Therefore the prediction model should be able to cope with both quantitative and categorical input variables.
- **Continuous output**: the output variable (manufacturing time) is a continuous variable. In order for the prediction model to be able to predict manufacturing times of new products, the output of the prediction model should be continuous.

Furthermore, a prerequisite non related to the data characteristics should be taken into account.

- **Interpretability**: since the construction industry is known to be conservative, the prediction model should be easily interpretable in order to increase the rate of acceptance in the industry.

3.3.1. Linear regression

Linear regression analysis is a technique used for making predictive models based on collected quantitative data. It is a relatively easy and flexible technique and is therefore widely accepted for making predictions [4]. When a linear relation in a dataset (x_1, y_1), (x_2, y_2),...,(x_n, y_n) is expected, a (simple) linear regression model can be used to make predictions for unseen data [15]. Linear regression aims at obtaining a linear function that represents the data accurately. The basic formula for linear regression is shown in equation 3.1. An example of a (visualized) linear regression model is provided in Figure 3.3. [15]

$$y = \alpha + \beta x \tag{3.1}$$

Where x is the predicting variable and y the response variable. α is called the intercept, and is the constant value of y if x equals zero. β represents the slope of the linear function.
In order to identify α and β, the method of *Ordinary Least Squares* (**OLS**) is used. This method aims at minimizing the total squared error between the predicted values and the measurements. The related function is provided in Equation 3.2 [15].

$$S(\alpha, \beta) = \arg\min \sum_{i=1}^{n} (y_i - \alpha - \beta x_i)^2 \tag{3.2}$$

In order to build a valid **MLR** prediction model using **OLS**, several assumptions should be met: [4].

- There is no multicollinearity between input variables
- The variance of errors is constant
- There is no autocorrelation between errors
- The errors are normally distributed

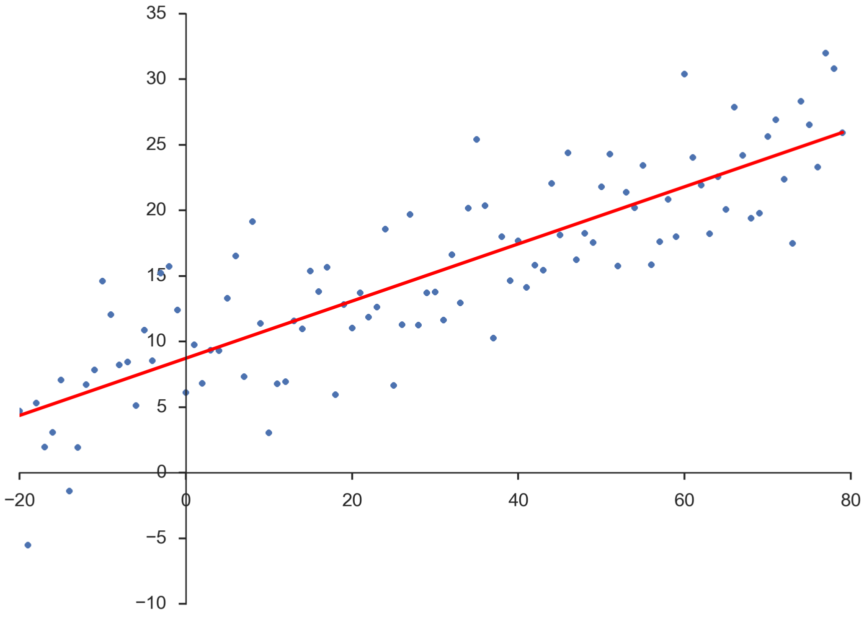


Figure 3.3: Example of (simple) linear regression [24]

While simple linear regression is limited to one input variable, multiple linear regression (**MLR**) takes into account more than one input variables. In **MLR** each predicting variable has its own slope coefficient β_j [20]. Implementing multiple input variables can result in more accurate prediction models. The standard formula for **MLR** is provided in equation 3.3, where Y is the predicted value, α is the intercept of the slope, $\beta_1,...\beta_k$ are the respective slopes of the input variables $x_1,...,x_k$.

$$Y = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \tag{3.3}$$

Analogue to simple linear regression, the parameters for the input variables can be determined using the least squares method (Equation 3.2). However, if multicollinearity exists between input variables, the least squares method becomes unstable [16]. Additionally, the least squares method often results in low bias, but high variance in the estimates [53]. An alternative exists in the Ridge regression approach, in which a little bias is sacrificed in order to reduce variance, which results in equation 3.4. Ridge regression uses a regularization parameter λ in combination with the variable slopes β_j, in order to minimize the slopes of the respective variables. The formula for Ridge regression is shown in equation 3.1. [16]

$$(\hat{\alpha},\hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 + \lambda \sum_j \beta_j^2 \right\} \tag{3.4}$$

The main challenge of using **MLR** for creating a prediction model, is selecting a proper set of input variables [40]. Implementing more input variables makes it possible to explain more of the variance in the output variable. On the other hand, minimizing the number of input variables reduces the chance of collinearity, over-fitting and transferring noise into the model [19].

The LASSO approach is based on the least squares method as shown in equation 3.2 and the Ridge Regression approach shown in equation 3.4. In contrast to Ridge Regression, LASSO could result in variable slopes β_j equal to zero, which implicates that the input variable is removed from the model. The resulting equation is shown in equation 3.5 [53].

$$(\hat{\alpha},\hat{\beta}) = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 + \lambda \sum_j |\beta_j| \right\} \tag{3.5}$$

With

$$t > 0, \quad \sum_{j=0}^{p} |\beta_j| < t \tag{3.6}$$

If a group of input variables is used, which have high pairwise correlations, LASSO selects one variable, which is not necessarily the best input variable. In a situation where the number of observations is significantly higher than the number of input variables, and high correlation exists between these variables, a Ridge model usually outperforms the LASSO approach. [59]

Zou and Hastie [59] therefore proposed the ElasticNET, in which Ridge and LASSO are combined, to create a model that performs well both with and without high correlation between inputs. The general equation of ElasticNET is provided in equation 3.7.

$$\hat{\beta} = \arg\min \left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 + \lambda_2 \sum_j |\beta_j| + \lambda_1 \sum_j \beta_j^2 \right\} \tag{3.7}$$

Models based on **MLR** provide a powerful and intuitive method to predict quantitative values. It is, however, limited to variables that have linear relationships and is inadequate to cope with categorical input variables.

3.3.2. Tree Based Regression

A different frequently used model to predict numerical values is Tree Based Regression (**TBR**). The construction of a regression tree is analogue to reasoning to a decision. Based on variable X_1, a (sub)decision is made. This process is repeated for all variables $X_1, X_2, ...X_i$, until the decision space is divided into a certain number of non-overlapping regions $R_1, R_2, ..., R_J$. For every observation that falls in region R_J, the prediction is equal to the mean of the training observations corresponding to region R_J [29]. A visualization of a regression tree is provided in Figure 3.4.

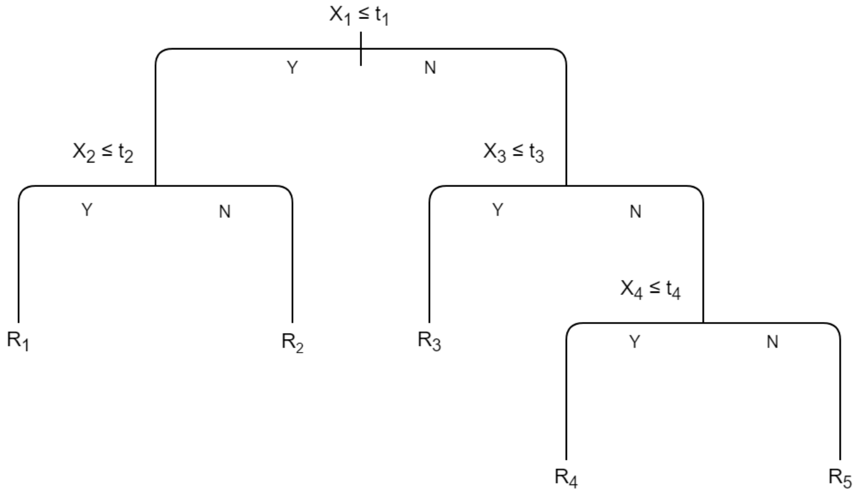


Figure 3.4: Example of a regression tree [29]

Finding decision, or split-criteria, that results in optimal predictions is proven to be NP-complete [28]. Therefore heuristics are commonly used to find near-optimal solutions for trees of growing size. Like in **OLS** (equation 3.2), the aim is to minimise the prediction error of the model. For a regression tree this leads to equation 3.8. [29]

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_j})^2 \tag{3.8}$$

With y_{R_j} being the mean response of training samples y_i within R_j. With a growing number of variables, the number of possible regions R_J increases significantly, resulting in infeasible computational times for determining the optimal set of regions. Therefore, a greedy approach is commonly used, in which the decision space is divided into two regions R_1, R_2, using a splitting value resulting in the greatest reduction in RSS (Equation 3.9). This process is repeated, dividing the previously identified regions, until a stopping criterion is met. Equation 3.9 shows the optimization used during the greedy search. The combination of input variable (j), with split value (s) resulting in the smallest prediction error between the real value y_i and the predicted value $\hat{y}_{R_1}$ is used to determine the next split. [29].

$$\sum_{i:x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:x_i \in R_{2(j,s)}} (y_i - \hat{y}_{R_2})^2 \tag{3.9}$$

Where

$$R_1(j,s) = \{X|X_j < s\} \qquad \text{and} \qquad R_2(j,s) = \{X|X_j \geq s\} \tag{3.10}$$

Using a **TBR** model for predicting a value based on its variables $X_1, X_2, ..., X_3$ is a convenient approach, since all decisions are insightful. Furthermore, **TBR** models allow for the implementation of discrete variables in the decision process. A major drawback, however, is that if a value results in a certain regio R, the resulting prediction is equal to the mean of the samples of region R. These discrete predictions lends itself less for the prediction of new data. Additionally, **TBR** models tend to overfit to the training data, leading to inaccurate predictions for new data.[29]

3.3.3. Linear Model Tree

As stated in section 3.3.2, Tree Based Regression (**TBR**) can be used to divide the dataset in smaller data sets. **TBR** has values at the leaves of the tree, equal to the average of the training data reaching the leaf. Quinlan [46] aims at combining the advantages of both **MLR** and **TBR** by proposing a Linear Model Tree (**LMT**). Like **TBR**, a tree based model is constructed based on input variables, reducing the variance of the sample set. The main difference is that rather than a value at the leaf, the leaves contain a **MLR** model. Consequently, predictions become a continuous, rather than discrete function, enhancing the prediction model to yield accurate predictions for new data [46]. A visualization of a **LMT** prediction model is provided in Figure 3.5.

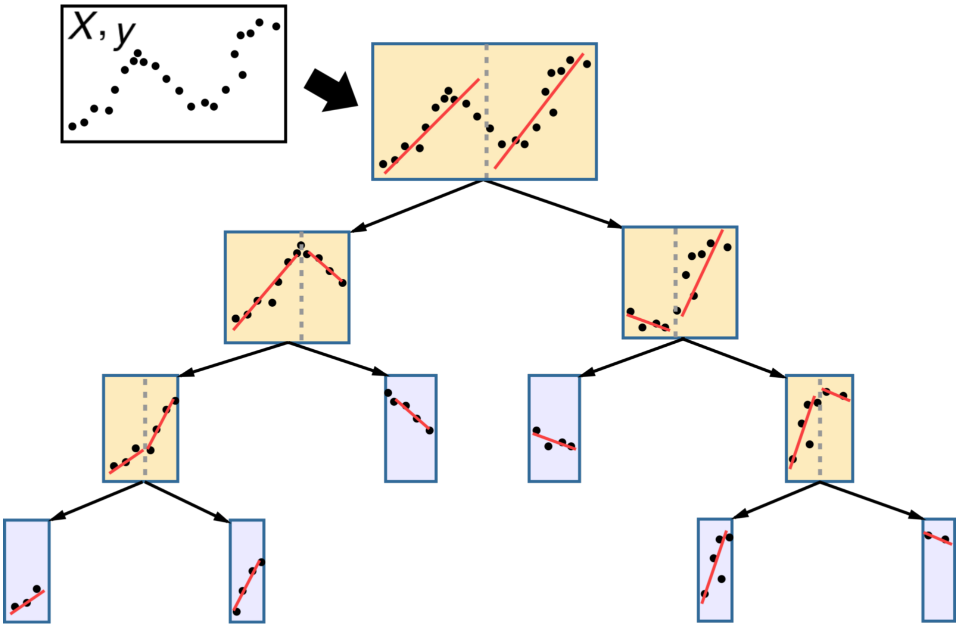


Figure 3.5: Schematic of a Linear Model Tree for a data set with one input variable[57]

The main advantages of a **LMT** prediction model, is that categorical input variables can be implemented in the model and that nonlinear trends can be divided into linear trends. Therefore, this approach is highly flexible; it should only split the data if nonlinear trends arise and can thus be used in both linear and nonlinear situations. Like **TBR** models, **LMT** models are troubled by overfitting problems. In addition, the **MLR** prediction models continue to be influenced by uncertainty in the data.

3.3.4. Support Vector Regression

Support Vector Regression (**SVR**) is a regression approach introduced by Drucker et al. [17]. Instead of aiming to find a line through the data points, like **MLR**, **SVR** attempts to find the narrowest tube around a function f(x). This tube has at most ϵ deviation from this function f(x). Furthermore, the sum of the distance of data points (ξ) deviating from this tube is minimized. This could be written as the optimization problem, described in equation 3.11 and equation 3.12. Here w are the slopes of the variables, and C is a regularization parameter. With increasing value C, the penalty on points outside the tube increases. The visualization of a model built using **SVR** is provided in Figure 3.6. [5]

$$\arg\min\left\{\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}\left(\xi_i+\xi_i^*\right)\right\} \tag{3.11}$$

Subject to

$$\begin{gathered} y_i - \{w, x_i\} - b \leq \epsilon + \xi_i \\ \{w, x_i\} + b - y_i \leq \epsilon + \xi_i \\ \xi_i, \xi_i^* \geq 0 \end{gathered} \tag{3.12}$$

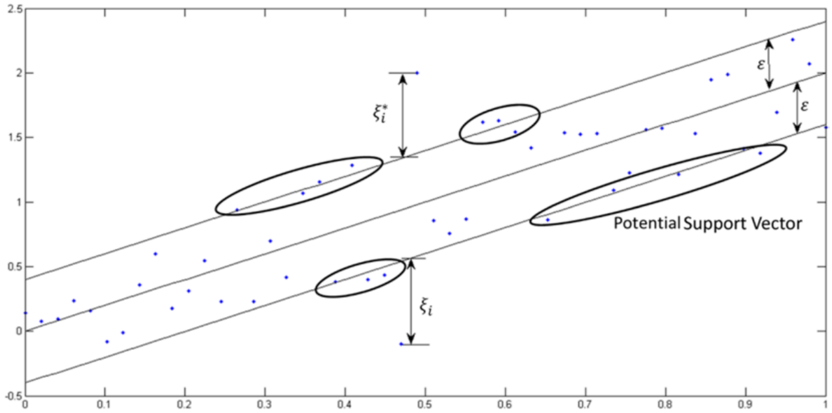


Figure 3.6: Visualization of Support Vector Regression [5]

Since each data point inside the tube does not influence the slope of the prediction model, **SVR** is less influenced by noise than **MLR**. **SVR** can be applied in both linear and nonlinear cases. For nonlinear cases, a trick is used where the data is mapped from a higher dimensional, or kernel space, to a linear space. This trick, however, requires significant trial and error based parameter optimization. In addition, the kernel space mapping trick comes with a reduction in interpretability of the **SVR** prediction model [51]. Similar to **MLR** prediction models, (basic) **SVR** prediction models are inadequate to cope with categorical input variables [17].

3.3.5. Neural Networks

Like the human brain, standard Neural Networks (**NN**) exists of many connected neurons. Neurons get activated upon an input, producing a sequence of activated neurons. Each neuron is weighted during the training process. Upon an input signal, the combined weights of the sequence of activated neurons provide the output the model. [50]

The training of a **NN** proceeds in two phases; the forward phase and the backward phase. In the forward phase, an input signal is transmitted through the network, going from layer to layer, until the output layer is reached. The output of the forward phase is compared by the desired response, and the resulting error is sent through the network once again, but this time the procedure goes from back to forth. The weights of the neurons are adjusted in this backward phase [25]. This process is shown in Figure 3.8.

Even though **NN**s tend to outperform other regression models in terms of prediction accuracy, the interpretability of these models is significantly more challenging than previously discussed models [25]. Additionally, **NN**s are relatively sensitive to uncertainty in the data [32].

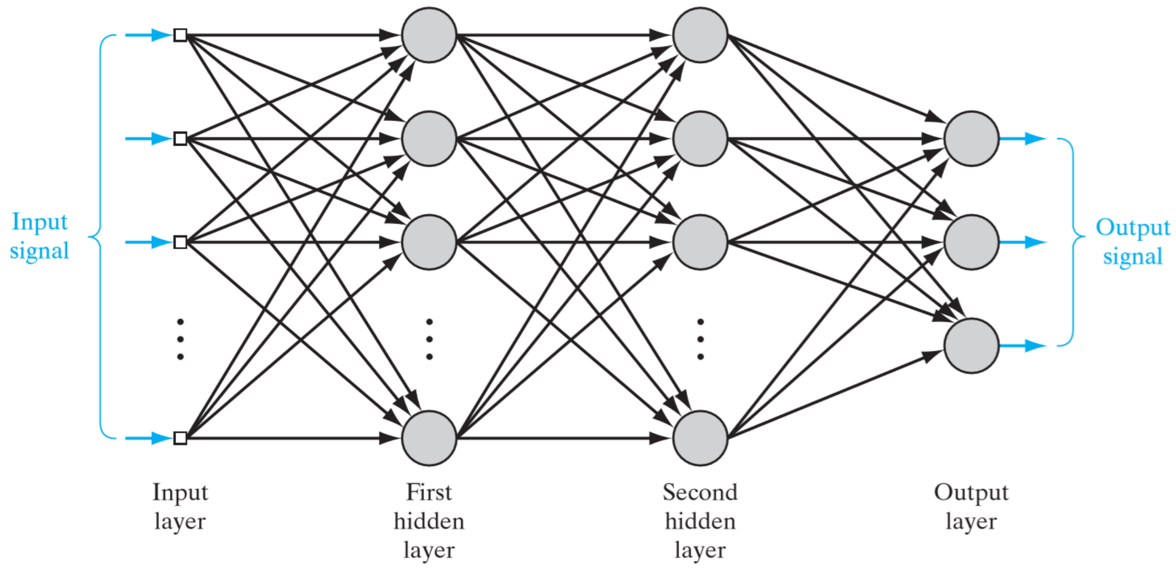


Figure 3.7: Visualization of a standard Neural Network [25]

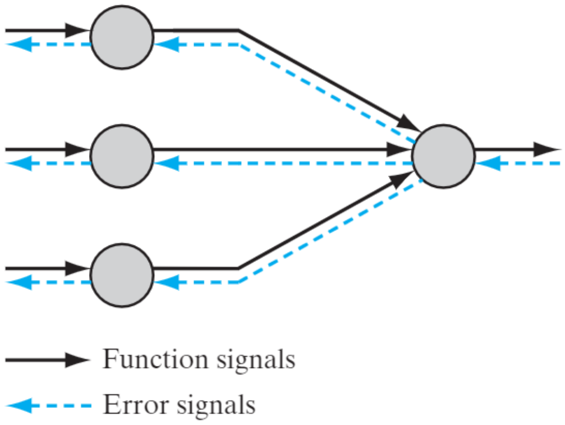


Figure 3.8: Visualization of the training process of a Neural Network [25]

3.3.6. Overview

In this section, identified prediction models are discussed in more detail. Based on the prerequisites derived in section 2.4.3, the different prediction models can be compared.

The **MLR** model would not provide a general solution for predicting manufacturing times using **BIM**, since it is only applicable in cases where the relation between physical properties and manufacturing times is linear. Moreover, it is impossible to implement categorical input variables in a **MLR** based model. Support Vector Regression (**SVR**) models have the same drawbacks as **MLR**, with the exception that **SVR** is less influenced by uncertainty in the data.

Tree Based Regression **TBR** overcomes the problem of nonlinear relationships in the data and is able to take categorical variables into account. **TBR**, however, results in discrete, rather than continuous prediction values.

The Linear Model Tree (**LMT**) approach combines advantages of both **MLR** and **TBR** models by constructing a regression tree with linear models in the nodes of the tree. Since **LMT** models can be used both in case of linear relationships (the root of the tree is a leaf) and nonlinear relationships (relationship is broken down to multiple linear relationships), **LMT** models have high general capabilities for predicting manufacturing times. Additionally, **LMT** models have high interpretability (especially compared with Neural Networks (NN)), since each splitting criterium can be visualized in the tree structure.

As discussed in section 3.3.5, Neural Networks (**NN**) tend to outperform more traditional methods like **MLR** and **SVR**, but comes with significantly lower interpretability.

Table 3.2 shows a comparison of the evaluated prediction models, based on the prerequisites for the prediction models to be used in this study. Considering generality, accuracy and interpretability of the evaluated prediction models, the **LMT** prediction model seems to be the most promising solution for this research. Therefore, the possibilities for implementing **LMT** models in order to predict manufacturing times using **BIM** will be further investigated.

The identified drawback of the **LMT** prediction model is that the prediction model is influenced by uncertainty in the data. As shown in section 3.3.1, the **MLR** prediction models in the nodes of the **LMT** model are influenced by noise in the data. An opportunity for improvement can be identified by replacing the **MLR** prediction models by **SVR** prediction models. **SVR** prediction models tend to outperform **MLR** models in case of uncertainty in the data. It is therefore expected that the the proposed adaptation to the **LMT** yields accurate predictions in datasets with significant uncertainty in the data as well. Since the proposed prediction model is a Model Tree with Support Vector Regression models in its nodes, the proposed model will be named Support Vector Regression Model Tree (**SVRMT**). For convenience, the **SVRMT** prediction model is added to table 3.2.

Table 3.2: Comparison of different prediction models based on set prerequisites

	Output function	**Input variables**	**Accurate for**	**Robust against uncertainty**	**Interpretability**
MLR	Continuous	Quantitative	Linear	No	Good
SVR	Continuous	Quantitative	Linear	Yes	Good
TBR	Discrete	Quantitative / Categorical	Linear / Nonlinear	No	Good
NN	Continuous	Quantitative / Categorical	Linear / Nonlinear	No	Bad
LMT	Continuous	Quantitative / Categorical	Linear / Nonlinear	No	Good
SVRMT	Continuous	Quantitative / Categorical	Linear / Nonlinear	Yes	Good

3.4. Conclusion

In this chapter, the state-of-the-art of predicting manufacturing times is discussed in order to answer the subquestion: *"Which manufacturing time prediction methods are available in literature?"*. Through a literature review, several studies on predicting manufacturing times have been identified. From this literature review, different prediction models are identified; Multiple Linear Regression (**MLR**), Tree Based Regression (**TBR**), Linear Model Tree (**LMT**), Support Vector Regression (**SVR**) and Neural Networks (**NN**). These studies, however, include only limited research focusing on the construction industry. In addition, these studies focus on predicting product lead time, rather than manufacturing time per manufacturing step.

After the various prediction models have been identified and explained in detail, the different prediction models are compared using the prerequisites derived in section 2.4.3. Based on this comparison, the Linear Model Tree (**LMT**) turned out to be the most promising prediction model for the prediction of manufacturing times per manufacturing step. An adaptation to this prediction model is proposed. In this adaptation, the **MLR** prediction models in the nodes of the Model Tree are replaced by **SVR** prediction models. It is expected that this adaptation will make the Support Vector Regression Model Tree (**SVRMT**) more robust to uncertainty in the data.

4
Development of the prediction model

4.1. Introduction

In section 3.3 an overview is provided of various possible models suitable for predicting manufacturing times using data from **BIM**. After these models are compared based on prerequisites derived in section 2.4.3, a combination of a Linear Model Tree (**LMT**) with Support Vector Regression (**SVR**) prediction models in the nodes of the tree is pointed out to be the (theoretically) most promising prediction model.

In this chapter, the development of the newly proposed Support Vector Regression Model Tree (**SVRMT**) will be discussed in detail. The construction of the **SVRMT** prediction model is based mostly on the construction of a **LMT** prediction model.

This chapter aims at answering the sub question: *"Which conceptual model can best be used to predict manufacturing times?"*.

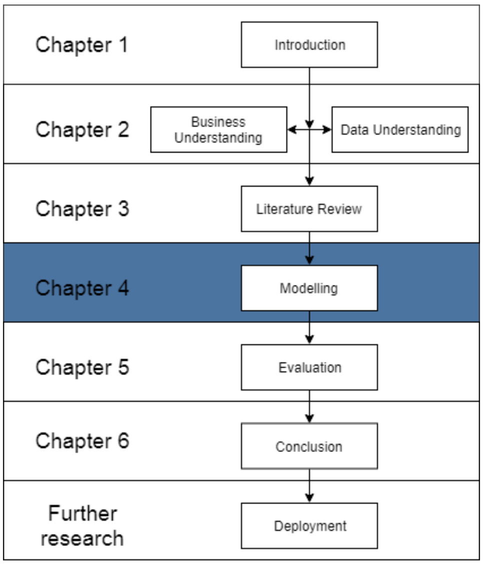


Figure 4.1: Fifth step of the used methodology for this research

4.2. Construction of a SVRMT

Even though the first mention of a model tree is provided by Quinlan [46], details of the prediction model were not given until Wang and Witten [56] extended the research on **LMT** models. Three steps can be identified in the construction process of a model tree; the splitting of nodes, creating a **MLR** model for each node and the pruning of the tree. The pseudo-code for the general construction of a model tree is provided in Algorithm 1.

Algorithm 1: Construction Model Tree

```
Construct Tree ;
    root = node ;
    Split(root) ;
    foreach node do
        model(node)
    foreach interior node do
        prune(node)
return Model Tree
```

4.2.1. Splitting an SVRMT

Since the splitting of an **SVRMT** prediction model is analogue to the splitting of a **LMT** prediction model, this section elaborates on the splitting of a **LMT** as described by Quinlan [46] and Wang and Witten [56]. As stated in section 3.3.2, finding an optimal regression tree is NP-complete. In order to find near-optimal decision trees, heuristics are used during the construction of a regression tree. The heuristic used for constructing a **LMT** varies slightly from the construction of a Regression Tree. During the construction of a **LMT**, the standard deviation of the target values Y is calculated. For each variable, the set is sorted. All possible split values of these variables are evaluated by calculating the standard deviations of the target values Y for both the left and right split set. The reduction in standard deviation by splitting the node is derived using equation 4.1. Using a greedy approach, the split leading to the biggest reduction in variance is chosen. The pseudo-code for splitting nodes to construct a **LMT** is shown in Algorithm 2. [46]

$$SDR = sd(T) - \sum_{i} \left(\frac{|T_i|}{|T|} * sd(T_i) \right) \tag{4.1}$$

With:

- sd(): the standard deviation of the set
- T: the total inherited subset
- T_i: the subset derived from the splitting value

This process is repeated until the node reaches a certain size (at least the number of input variables), or until splitting the node does not result in a reduction of standard deviation larger than 5% of the standard deviation of the parental node . When this criteria is met, the node is called a leaf. [56]

The heuristic of the original **LMT** splitting procedure considers the reduction of standard deviation in target values (Y) that results from splitting the node. As stated by Karalič [31], this heuristic is not an appropriate measure. Perfectly linear data can have a large standard deviation and would thus be split according to the heuristic used in the original approach, even though the data can perfectly be described with a linear model.

Karalič [31] therefore proposes a heuristic that for each split a simple linear regression model using the split variable is built for both the left and right child. The residuals are derived using the simple linear regression model and the reduction in standard deviation of the residuals is maximized.

This approach can be extended by building a **MLR** model using multiple input variables after each split. Using **MLR** models instead of simple linear regression models is especially useful in case a split based on a categorical input variable is considered. As stated in section 3.3.1 simple linear regression is unable to incorporate categorical input variables. Therefore, a categorical input variable is less likely to be chosen over a quantitative input variable as split variable. The major drawback of using **MLR** over simple linear regression is that the computational complexity grows significantly with the number of input variables [31].

Algorithm 2: Split Model Tree

```
Data: Node
Result: Split Node
foreach predictor variable do
    foreach possible split value do
        Split data ;
        if size(left data) > MinSize AND size(right data > MinSize) then
            Calculate SDR ;
if Improvement found then
    node.attribute = predictor variable leading to maximum SDR ;
    node.splitvalue = value leading to maximum SDR ;
    node.type = interior;
    create left child ;
    create right child ;
    split(left child);
    split(right child);
else
    node.type = leaf
```

An example of a **LMT** prediction model is provided in Figure 4.2. In this example, a quadratic relationship between input (X) and output (Y) is split in multiple linear segments.

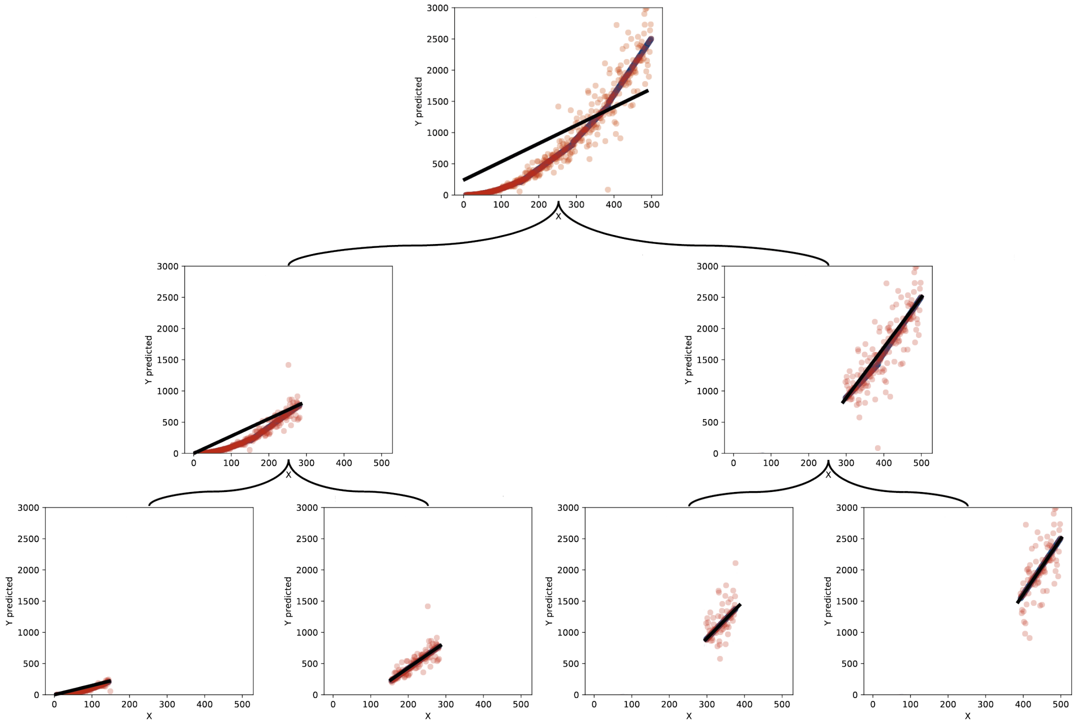


Figure 4.2: Schematic of a Linear Model Tree for a data set with one input variable

4.2.2. Node model

In the original **LMT** prediction model, a linear model is derived for each node. This could be either a simple linear regression model, or an **MLR**. The approaches described in section 3.3.1 can be used for this step.

As discussed in section 3.3.4, **SVR** based models tend to outperform **MLR** models when uncertainty in the dataset is present. The data analysis in section 2.4 shows that uncertainty is present in the dataset of this case study. Since this uncertainty is human related, it is expected that the presence of uncertainty will occur in all construction industry related manufacturing processes. Therefore, in this study, Support Vector Regression (**SVR**) models are suggested for the nodes. [5]

For the implementation of a **SVR** prediction model in each node of the tree, equation 4.2 is used.

$$\arg\min \left\{ \frac{1}{2} ||w||^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \right\} \tag{4.2}$$

Subject to

$$\begin{aligned} y_i - \{w, x_i\} - b &\leq \epsilon + \xi_i \\ \{w, x_i\} + b - y_i &\leq \epsilon + \xi_i \\ \xi_i, \xi_i^* &\geq 0 \end{aligned} \tag{4.3}$$

In order to optimize a **SVR** prediction model, two parameters should be tuned. The value "C", which determines the penalty for points not in the "tube" of the **SVR** prediction model. Additionally, the parameter ϵ is used to determine the width of the "tube" in which points are not penalized in the optimization function. Commonly, parameter tuning is performed using grid search. During grid search, several possible combinations for C and ϵ are tested by splitting the data in a train and test set [12].

The grid search approach is performed through k-fold validation. K-fold validation splits the training data in a training and validation fold in order to prevent overfitting to the training data. The size of both training and validation fold is based on the value of k; the training fold covers a $\frac{k-1}{k}$ portion of the original training data, while the validation covers $\frac{1}{k}$ of the training data. After this split, the prediction model is trained using the training fold and tested for the validation fold. This is repeated k times. Resulting from this strategy, each data point is tested once, eliminating 'lucky guesses'. Upon completing all k tests, the results are averaged. This k-fold validation is repeated for each possible combination of the grid search approach. A visualization of 5-fold validation is provided in Figure 4.3. [34]

The combination of C and ϵ resulting in most accurate predictions is chosen. This approach is straight forward, but a consideration between finding the best near optimal combination and computational costs should be made. [12]

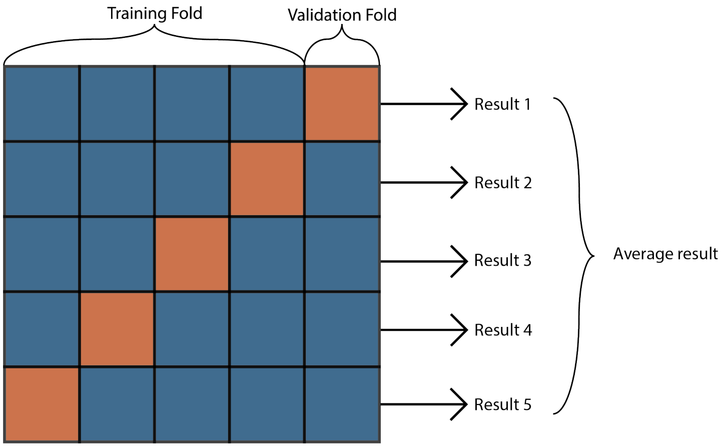


Figure 4.3: Visualization of 5-fold cross validation

Additionally, **SVR** prediction models have no feature selection methods embedded, therefore an alternate approach should be taken to select an appropriate subset of input variables. A backward elimination feature selection strategy will be used. This way input variables are dropped greedily, as long as they are not expected to improve the models accuracy. This strategy is insightful, computational advantageous and robust against overfitting [22].

Figure 4.4 shows a constructed **SVRMT** prediction model. Notable are the differences in the nodes of the model tree. Rather than a **MLR** model (a line through the points) a **SVR** model (smallest tube around the flattest line) is placed in the nodes of the model tree.

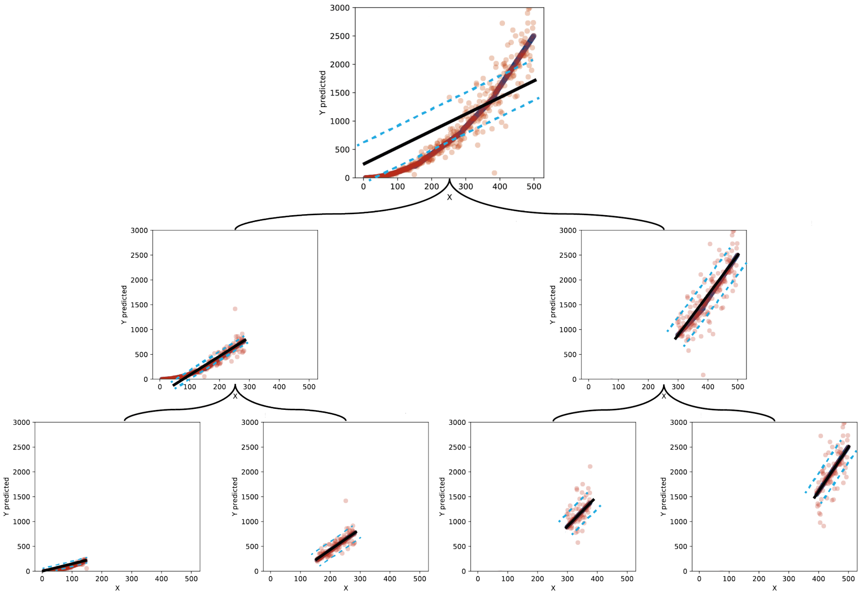


Figure 4.4: Schematic of a Support Vector Regression Model Tree for a data set with one input variable.

4.2.3. Pruning a model tree

Like **TBR**, chance of overfitting arises for a **LMT**. In order to prevent this overfitting, Quinlan [46] proposes a pruning method. After a prediction model is in place for each node, the tree is pruned. From the leaves to the root, all interior nodes are tested. In this test, the mean of the absolute residuals for the data points reaching the node is derived and multiplied by the factor of equation 4.4. This process is conducted for both child nodes as well. If the error term of the parental node is smaller than the combined expected error of the children nodes, the parental node is 'pruned' and becomes a leaf. The pseudo-code for this process is shown in Algorithm 3

Equation 4.4 is used for the computation of the expected error due to overfitting. With n the size of the training set and v the number of input variables, y the real value and $\hat{y}$ the predicted value.

$$Error = \frac{n+v}{n-v} * \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.4}$$

Algorithm 3: Prune

```
Data: Interior node
calculate mean(residuals) of node;
use equation 4.4 to determine factor;
error = factor*mean(residuals);
foreach child node do
    calculate mean(residuals of node ;
    use equation 4.4 to determine factor;
    error = factor*mean(residuals);
if error(node) < $\sum_i \left(\frac{|T_i|}{|T|} * error_i\right)$ then
    node.type = leaf;
else
node.type = interior;
```

4.3. Predicting using a SVRMT

In order to predict a new value using the **SVRMT** prediction model, the **SVRMT** is descended using the decision criteria of the **SVRMT** until the matching leaf is reached. After the corresponding leaf has been found, the **SVR** prediction model at the leaf is used to compute the predicted value. The prediction is smoothed by ascending the tree from leaf to root. At each node along the path to the root, the predicted value is smoothed using equation 4.5 to compensate for possible sharp discontinuities between adjacent leaves [46]. Where n equals the number of training data points reaching the child node, p is the predicted value passed from the child node, q is the predicted value using the linear model from the current node and k is a smoothing parameter (Quinlan [46] suggests a default value 15 for k).

$$p' = \frac{np + kq}{n + k} \tag{4.5}$$

Algorithm 4: Smoothing Model Tree

```
Data: Leaf
Result: Smoothed prediction
while node is not root do
    if node is leaf then
        p = node.predict ;
        node = node.parent ;
    else
        n = size(node.child) ;
        q = node.predict;
        calculate p using equation 4.5;
        node = node.parent;
n = size(node.child) ;
q = node.predict;
calculate p using equation 4.5;
return p
```

4.4. Conclusion

Upon the comparison of different available prediction models in section 3.3.6, this chapter provides a detailed description of the (theoretical) development of the most promising prediction model (Support Vector Regression Model Tree (**SVRMT**) in order to answer the sub-question: *"Which conceptual model can best be used to predict manufacturing times?"*.

Basically, the construction of an **SVRMT** consists of a splitting, modelling and pruning phase. In order to compensate for sharp discontinuities between adjacent leaves, the predictions are smoothed from leaf to root. In order to be more robust against uncertainty in the data, it is proposed to use Support Vector Regression (**SVR**) models in the nodes of a Linear Model Tree, instead of Multiple Linear Regression (**MLR**) models. For the tuning of these **SVR** prediction models, grid-search in combination with 5-fold validation is used to find the (near) optimal combination of parameters.

5
Model Validation

5.1. Introduction

After the development of the proposed Support Vector Regression Model Tree (**SVRMT**) has been discussed in detail (chapter 4), the proposed prediction model is tested. Through a series of experiments, the sub-question *"How can the conceptual prediction model be validated?"* is aimed to be answered in this chapter.

The validation of the conceptual prediction model is divided in two phases. In the first phase, the assumptions leading to the proposal of the **SVRMT** prediction model are verified in hypothetical situations. During the second phase, the proposed **SVRMT** prediction model is validated in real case scenarios. For these real case scenarios, data from Oostingh Staalbouw is used.

Prior to conducting the validation experiments, performance indicators for prediction models are identified. Using these performance indicators, the (different) prediction model(s) can be evaluated in terms of prediction accuracy. This leads to answering the sub-question: *"How can the performance of prediction models be evaluated and compared?"*

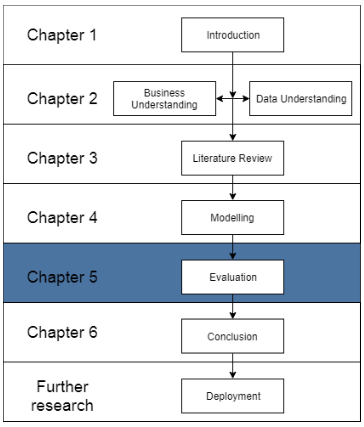


Figure 5.1: Sixth step of the used methodology for this research

In the experimental phase of this research, the performance of the **SVRMT** prediction model is compared with a Multiple Linear Regression (**MLR**), Support Vector Regression (**SVR**), Tree Based Regression (**TBR**) and standard Linear Model Tree (**LMT**) prediction model. Unless stated otherwise, for each experiment discussed in this chapter, the available data is split in a training and test set using a 70/30 ratio and is repeated tenfold.

For the implementation of the theoretical prediction model, Python 3.7 is used. Specifically, the Anaconda distribution [3] is used due to the availability of data analysis libraries. The Python code used for this research is provided in Appendix D.

5.2. Evaluation of prediction models

In order to evaluate the performance of prediction models in terms of prediction accuracy, several common methods are identified. These methods can be divided into metrical (Performance Indicators) and visual (Indicator Plots) methods. This section highlights commonly used methods for evaluating and comparing different prediction models.

5.2.1. Performance Indicators

Coefficient of determination

One numerical way to express the fitness of a relationship between predicted value and the real value is the implementation of the coefficient of determination (R^2, Equation 5.1). R^2 is a measure of the variance of Y explained by the model, where $R^2 = 0$ implicates that none of the variance is explained by the model, and $R^2 = 1$ suggests that all variance can be explained by the model. [48]

$$R^2 = 1 - \frac{\sum(\hat{Y} - Y)^2}{\sum(Y - \overline{Y})^2} \tag{5.1}$$

Where

- Y: actual value of Y
- $\hat{Y}$: predicted value of Y
- $\overline{Y}$: average value of Y

A model with R^2 close to 1 therefore suggests that the prediction model performs well in terms of prediction accuracy. However, a model with R^2 close to 1 can also indicate that the model is overfitted; the model is adjusted to noise in the training data and provides poor predictions on new data. A visualized example of overfitting is provide in Figure 5.2.

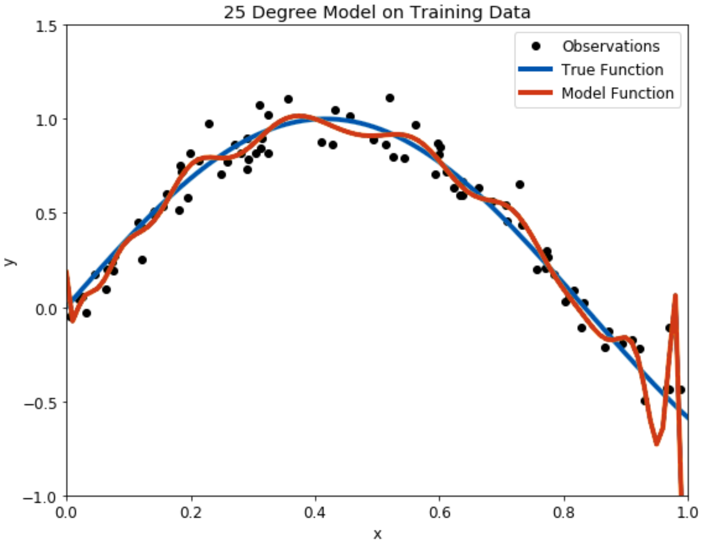


Figure 5.2: Visual example of an overfitted prediction model [33]

RMSE

Another widely adapted performance indicator for determining the accuracy of a prediction is the Root Mean Square Error (**RMSE**), in which the average prediction error is considered (Equation 5.2). **RMSE** can be used to evaluate the quality of different prediction models, since the objective is to minimize the prediction error. The statistic, however, is not effective for evaluating models across different datasets. Large values y, tend to have larger absolute errors leading to increased **RMSE**. The formula for deriving the **RMSE** is provided in equation 5.2, where y_i is the actual value and $\hat{y}_i$ is the predicted value. [23]

$$RMSE = \frac{\sqrt{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}}{n} \tag{5.2}$$

Since this Performance Indicator has limited usage to compare results between different data sets, the **RMSE** Performance Indicator will not be used in this research.

MAPE

Where the main shortcoming of the **RMSE** perfomance indicator is that it is ineffective to compare the performance of a prediction model across different data sets, the Mean Absolute Percentage Error (**MAPE**) offers a solution to this problem. The **MAPE** takes the absolute error in order to tackle the problem of positive and negative errors cancelling each other out. Since the performance indicator uses the relative absolute error, this performance indicator shows the relative accuracy of the prediction model. Therefore, this performance indicator is suitable to compare the accuracy of a prediction model across multiple datasets. Equation 5.3 is used to compute the **MAPE** performance indicator. [42]

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)/\hat{y}_i| \tag{5.3}$$

The range of **MAPE** is [0, +∞], with 0 meaning the predicted values are equal to the real values. A **MAPE** of 1 implying that the absolute error is equal to the real value (100% absolute relative error) and values greater than 1 implying an absolute error greater than the real value.

Additionally, the Median of the Absolute Percentage Error (**MeAPE**) and Standard Deviation of the Absolute Percentage Error (**Std**) can be used to gain further insight in the distribution of the absolute percentage errors.

5.2.2. Indicator plots

Alongside performance indicators, the performance of prediction models can be evaluated visually. Real output variables (Y) can be plotted against the predicted output variables ($\hat{Y}$). If the plotted points are close to a line under 45 degrees, this indicates that the prediction model is close to reality. These plots are called inverse fitted value plots [48]. An example of an inverse fitted value plot is shown in Figure 5.3.

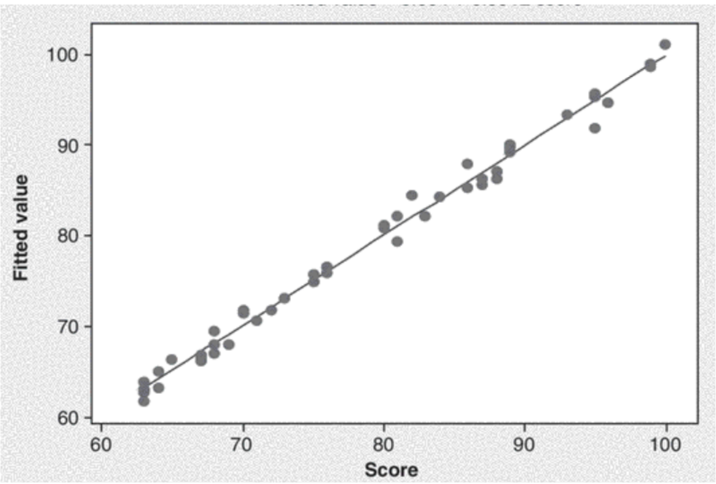


Figure 5.3: Example of an inverse fitted value plot. As the scatter points are close to a line under 45 degrees, this plot implies that the prediction model is close to reality [48].

The performance of different prediction models can be visually compared using boxplots. By generating a boxplot of the (absolute relative) residuals, the distribution of these residuals is shown visually. The mean, median and standard deviation are shown in these boxplots. Plotting boxplots of different prediction models next to each other provides (visual) insight in the relative performance of the different prediction models.

An example of a boxplot of absolute relative residuals is provided in Figure 5.4. In this example, interesting parts of the boxplot are highlighted.

- **Median**: The middle value of the dataset.
- **Q1**: This represents the middle number between the smallest number and the median of the dataset.
- **Mean**: The mean value of the dataset (**MAPE**), represented by the diamond shape in the Figure.
- **Q3**: This represents the middle number between the median and largest number of the dataset.
- **IQR**: The difference between Q1 and Q3.
- **Q1-1.5*IQR (Minimum)**: Represents the lower whisker of the boxplot, points lower than this minimum are considered outliers.
- **Q3+1.5*IQR (Maximum)**: Represents the upper whisker of the boxplot, points higher than this minimum are considered outliers.

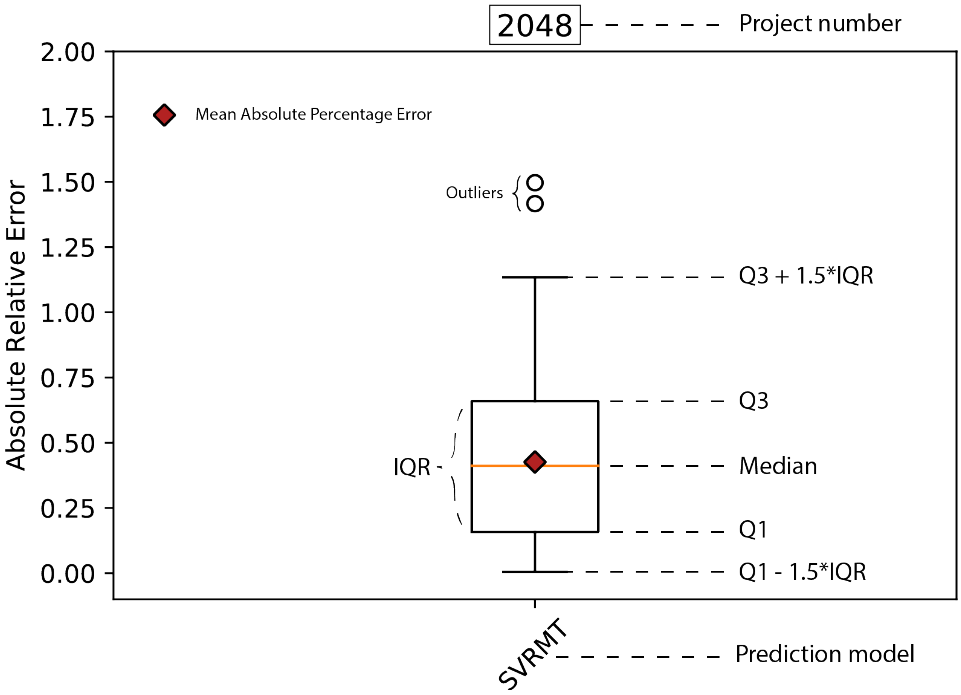


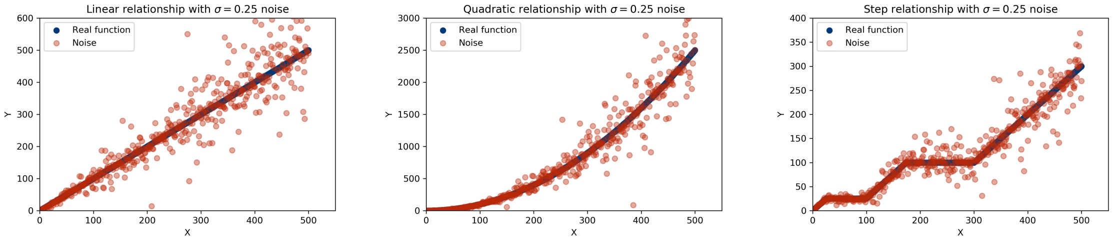
Figure 5.4: Example of a boxplot of absolute relative residuals used in the second experimental phase of this research

5.3. Phase I: Verification

Based on the reasoning of chapter 4, the proposed Support Vector Regression Model Tree (**SVRMT**) should be able to deal with both linear and nonlinear relationships. In addition, the proposed prediction model should be robust against influence from uncertainty in the data. In order to verify these assumptions, several hypothetical relationships are evaluated:

1. **Linear**: This relationship is tested in order to verify the assumption that the proposed prediction model has comparable accuracy with linear prediction models.
2. **Quadratic**: This relationship is regarded to check the assumption that the proposed model is able to yield accurate predictions for nonlinear relationships.
3. **Step**: In order to verify the assumption that both quantitative and categorical predictor variables (X) can be taken into account by the proposed model, a step relationship is regarded.

The robustness against uncertainty in the data is verified by adding heteroskedastic noise to the relationships. The dependent variable (Y) of the relationship is multiplied by a normal distribution. For this normal distribution μ is 0 and σ = 0.25 are used, which is in line with the standard deviation of manufacturing times for identical elements manufactured at Oostingh Staalbouw, discussed in chapter 2.4 (table 2.5). In addition, random outliers are added to the noise.



(a) (b) (c)

Figure 5.5: Plots of relationships used for verification. Respectively a) linear b) quadratic and c) the step relationship

For the **MAPE**, **MeAPE** and **Std**, a small alteration is made compared to the definition provided in section 5.2.1. In order to evaluate the ability of the prediction models to reconstruct the relationship without added noise, the predicted relationship is compared with the real relationship, rather than the added noise. In the remainder of this section, notable results are highlighted. For a complete overview of results of the verification phase, the reader is directed to Appendix B.

5.3.1. Linear relationship

The first verification test is based on a linear relationship between the input (X) and output (Y) variable. This experiment is used to verify the assumption that the proposed **SVRMT** prediction model has comparable performance to linear models like **MLR** and **SVR**.

In table 5.1 the results of the verification experiment for the linear relationship are provided. The **SVR**, **LMT** and **SVRMT** model outperform the **MLR** model slightly in terms of prediction accuracy. This strengthens the expectation that the **MLR** is influenced more by uncertainty in the data than the **SVR** prediction model. As expected, the **SVR** and **SVRMT** model yield equal results, which indicates that the **SVRMT** consists solely of a root with a **SVR** model in it.

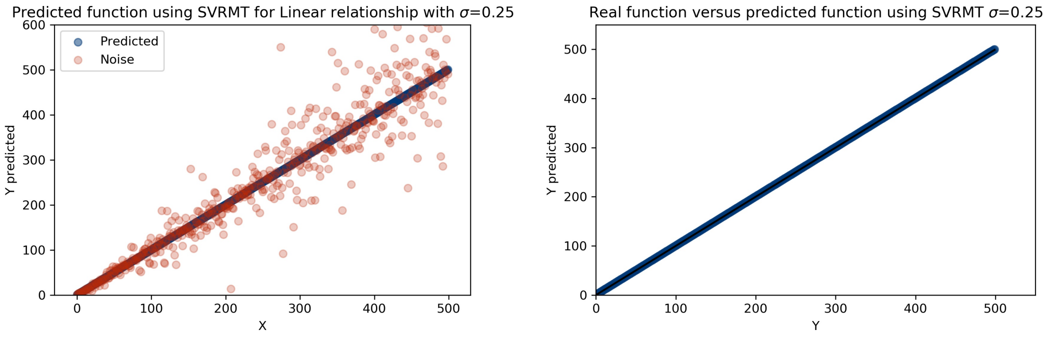
Table 5.1: Results of the evaluated prediction models for a linear relationship

	MAPE	**MeAPE**	**Std**	**R^2**
MLR	0.06	0.03	0.02	0.84
SVR	0	0	0.02	0.99
TBR	0.10	0.05	0.35	0.83
LMT	0.04	0.02	0.07	0.88
SVRMT	0	0	0.02	0.99

The reconstruction of the linear relationship by the **SVRMT** prediction model is shown in Figure 5.6a. In order to compare the reconstructed relationship with the real relationship, Figure 5.6b shows the predicted value versus the real value (relationship without noise added). As the plotted points lie on a line under 45 degrees with the horizontal axis, this Figure indicates that the **SVRMT** prediction model is capable of accurately reconstructing the linear relationship from the noisy data.

From table 5.1 the **MAPE** of the **LMT** prediction model suggests that this model is capable of reconstructing the linear relationship from the noisy data as well. Figure 5.7a shows that the **LMT** model is broken down into several linear segments, with varying slopes. This implies that the **LMT** model is influenced by the noise in the data and is getting trapped in local minima.

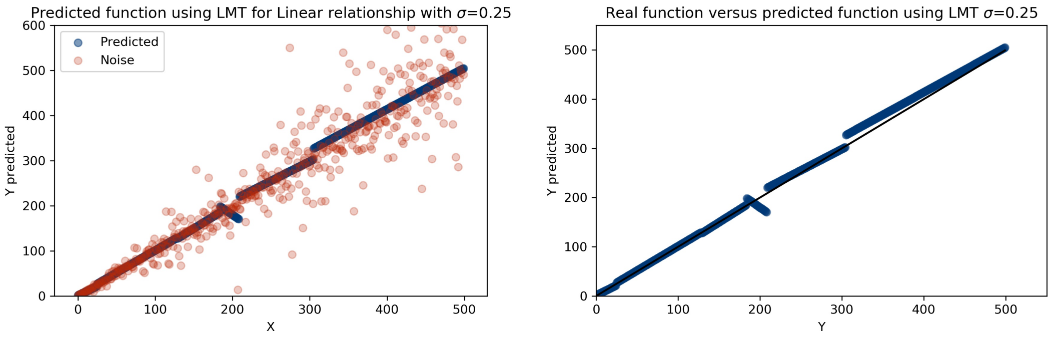
The difference between **LMT** and **SVRMT** models can be explained by the fact that the splitting procedure of both the **LMT** and **SVRMT** model is done by evaluating a **MLR** model for both the left and right side of a split. Since **MLR** is known to be easily influenced by uncertainty in the data (section 3.3.1) both the **LMT** and **SVRMT** will overfit to the noise in the data. The **SVRMT**, however, creates **SVR** models in its leaves, which are



(a) (b)

Figure 5.6: a) Reconstructed linear relationship by the SVRMT prediction model b)inverse fitted value plot linear relationship SVRMT prediction model

known to be less influenced by uncertainty in the data than **MLR**. Therefore, in case of a linear relationship with significant uncertainty in the data, the **SVRMT** model will be pruned significantly more than the **LMT** prediction model. As a result, the **SVRMT** model is less prone to overfitting to noise than the **LMT** model.



(a) (b)

Figure 5.7: a) Reconstructed linear relationship by the LMT prediction model b)inverse fitted value plot linear relationship LMT prediction model

5.3.2. Nonlinear relationship

After the assumption that the **SVRMT** prediction model yields accurate results on a linear relationship is verified, nonlinear relationships are tested. In order to verify this assumption, two different nonlinear relationships are tested; a quadratic and a step relationship.

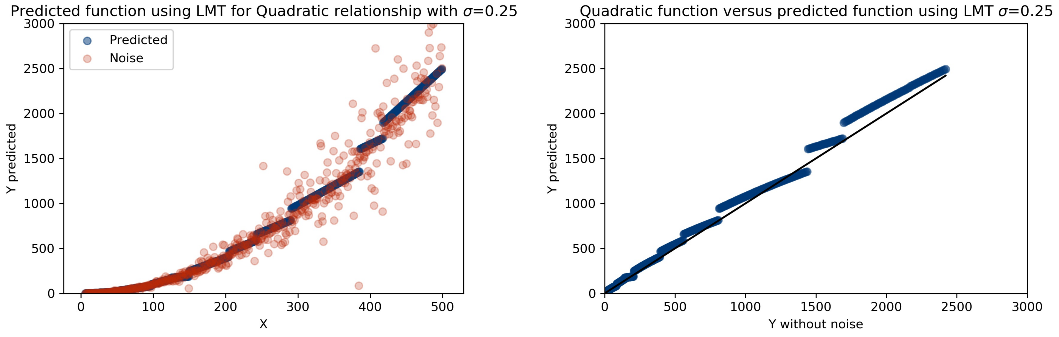
Quadratic

For the purpose of verifying the assumption that the proposed **SVRMT** model is able to predict nonlinear relationships, an experiment using a quadratic relationship is conducted. Analogue to the verification experiment for linear relationships, heteroskedastic noise is added to the relationship.

The Performance Indicators, shown in Table 5.2, show a minor difference between the **LMT** and **SVRMT** model. The reconstructed quadratic relationships by the **LMT** and **SVRMT** models are shown in respectively Figure 5.8a and Figure 5.9a. Analogue to the experiment with the linear relationship, the **LMT** model is influenced significantly more by uncertainty in the data compared to the **SVRMT** model.

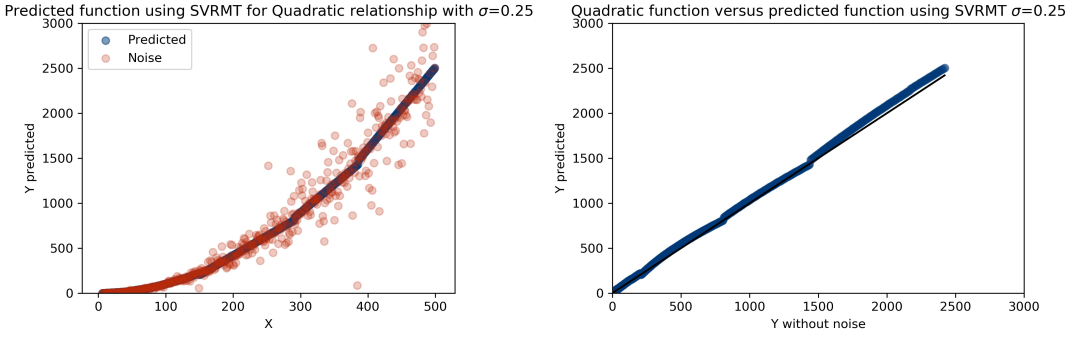
Table 5.2: Results of the evaluated prediction models for a quadratic relationship

	MAPE	MeAPE	Std	R^2
MLR	9.24	0.41	43.95	0.59
SVR	7.15	0.24	41.93	0.80
TBR	0.36	0.07	1.84	0.87
LMT	0.10	0.06	0.34	0.91
SVRMT	0.06	0.02	0.34	0.91



(a) (b)

Figure 5.8: a) Reconstructed quadratic relationship by the LMT prediction model b)inverse fitted value plot quadratic relationship LMT prediction model



(a) (b)

Figure 5.9: a) Reconstructed quadratic relationship by the SVRMT prediction model b)inverse fitted value plot quadratic relationship SVRMT prediction model

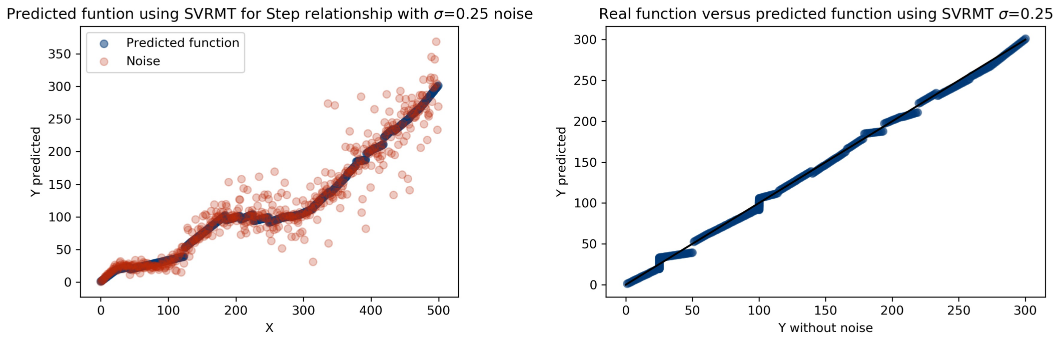
Step

Last, a relationship with two input variables is evaluated; one quantitative and one categorical variable. The relationship is linearly increasing if the categorical variable equals 1. If the categorical variable is 0, the relationship is constant, resulting in a step like function as shown in shown in Figure 5.5c.

The results shown in table 5.3, Figure 5.10a and Figure 5.10b show that the **SVRMT** is capable of reconstructing the step function with a small error. This error mostly occurs due to the combination of sharp discontinuities going from a constant relationship to a linear relationship along with the presence of noise.

Table 5.3: Results of the evaluated prediction models for a step relationship

	MAPE	MeAPE	Std	R^2
MLR	0.22	0.16	0.21	0.84
SVR	0.24	0,16	0.37	0.85
TBR	0.17	0.10	0.31	0.89
LMT	0.13	0.05	0.24	0.91
SVRMT	0.11	0.02	0.34	0.91


Predicted funtion using SVRMT for Step relationship with σ=0.25 noise

Real function versus predicted function using SVRMT σ=0.25

(a)

(b)

Figure 5.10: a) Reconstructed step relationship by the SVRMT prediction model b)inverse fitted value plot step relationship SVRMT prediction model

5.4. Phase II: Case study experiments

After the proposed prediction model is verified in several hypothetical scenarios, a case study is performed with data from the manufacturing process of Oostingh Staalbouw. During this case study, the accuracy of the proposed manufacturing time prediction model for real construction projects is evaluated. The proposed prediction model (**SVRMT**) along with **MLR**, **SVR**, **TBR** and **LMT** prediction models are tested in three different real case scenarios:

1. **Leave one project out**; In this scenario, all but one project are merged and used to train a prediction model. This prediction model is used to predict manufacturing times of products for a new project. This scenario represents the start of the manufacturing phase of a new project. Since no prior manufacturing data of the new project is available at this stage of the project, historical data from realized projects is the only suitable reference to predict the manufacturing times of products for this new project. Since the dataset is split based on a project, leaving one project out, there is no random split used. Therefore, no repetitional experiments are required for this scenario.

2. **Project partly manufactured**; This scenario corresponds to a stage where the manufacturing of the project has been in progress for a considerable period. In this stage, historical manufacturing data of the evaluated construction project becomes available. This historical data can be used to predict the manufacturing times for the remainder of the construction project. Especially in case of relatively large construction projects, a considerable amount of data can be gathered. The expectation is that this data can be used to enhance the predictions of manufacturing times for the remaining products of the project.

3. **Combined scenario**; This scenario corresponds to the manufacturing stage of a construction project evaluated in scenario 2. Scenario 3, however, differs in that it uses both data from realized construction projects (analogue to scenario 1) along with gathered data from the specific construction project (scenario 2).

The different scenarios are visualized in Figure 5.11.

It is expected that the accuracy of the predictions will increase from scenario 1 through scenario 3 due to the increasing availability of relevant data. From the uncertainty in manufacturing times of repetitive manufactured products discussed in section 2.3.2 an objective **MAPE** of 0.30 will be used.

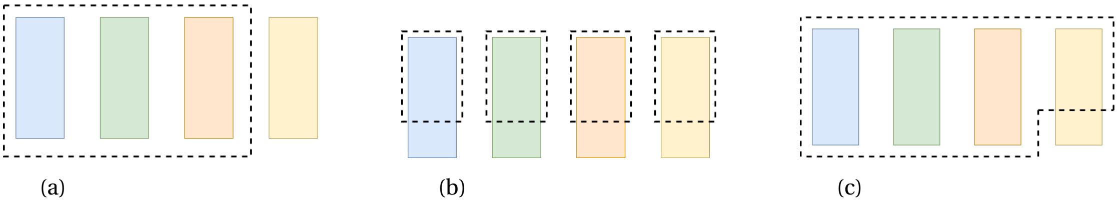


Figure 5.11: Visualization of a) scenario 1, b) scenario 2, and c) scenario 3. Each colored rectangle represent a different construction project. The available training data is represented by the dashed rectangle.

5.4.1. Projects evaluated for real case validation

For these experiments, four projects carried out at Oostingh Staalbouw are evaluated. Three of these four projects are the main projects carried out by the company in the last year (September 2018 - June 2019), while one of them is manufactured in 2017 (project 2048). The latter project is taken into account to check whether projects realized in previous years are still relevant for the prediction of new projects.

- **Project 2048**: A relatively small project with a large variety of products. This project is manufactured in 2017
- **Project 181017**: This project is a relatively large, industrial construction. This project took a significant part of the evaluated period to be manufactured.
- **Project 184062**: This is a medium sized project with only a small number of repetitive products.
- **Project 194003**: This is a relatively small project consisting mainly of repetitive products. Additionally, the physical properties of the products differ relatively more from the other evaluated projects.

A summary of properties of these projects, along with a visualization of the Building Information Models is provided in Appendix C.1. Data of the products manufactured for these projects are used in order to validate the proposed prediction model.

5.4.2. Scenario 1 - Leave one project out

For the first scenario, one project is left out from the training data. After the prediction model is trained using data from the other projects, the model is used to predict the manufacturing time of products for the project left out. This experiment is repeated for each project left out once. This scenario corresponds to the start of the manufacturing of a new project. Since no manufacturing data of the project left out is included in the prediction models, it is expected that the accuracy of the various prediction models is lowest in this scenario.

In table 5.4, the **MAPE** of the different prediction models for the experiments conducted for scenario 1 are shown. Notably are the differences between the linear models (**MLR** and **SVR**) compared to the nonlinear (**LMT** and **SVRMT**) models for the prediction of assembly times. Both **LMT** and **SVRMT** prediction model result in more accurate predictions, suggesting that the relationship between physical properties and assembly time is nonlinear. For the welding step, on the other hand, the small difference in terms of prediction accuracy between the **SVR** and **SVRMT** implies that the relationship between physical properties and welding time is linear.

Additionally, it is notable that the prediction of assembly times for project 194003 is slightly less accurate compared to the other projects. This difference can be explained by the fact that project 194003 differs significantly compared to the other projects in terms of physical properties.

For both the prediction of assembly and welding times, the proposed **SVRMT** yields both most constant and accurate results in terms of prediction error.

Table 5.4: MAPE of the results for the different implemented prediction models in scenario 1 for predicting the assembly (left) and welding (right) times. The bold numbers represent the lowest MAPE of the evaluated prediction models per project.

	2048	181017	184062	194003	Average
MLR	0.45	0.61	0.52	1.36	0.74
SVR	0.45	0.56	0.34	1.15	0.63
TBR	0.72	0.91	0.42	0.57	0.66
LMT	**0.43**	0.44	0.46	0.56	0.47
SVRMT	**0.43**	**0.41**	**0.34**	**0.45**	**0.41**

	2048	181017	184062	194003	Average
MLR	0.45	0.36	0.51	0.41	0.43
SVR	0.44	**0.34**	**0.42**	0.40	**0.40**
TBR	0.53	0.62	0.63	0.38	0.54
LMT	**0.41**	0.36	0.56	**0.43**	0.44
SVRMT	0.42	0.36	**0.42**	0.41	**0.40**

In order to compare the different implemented models visually, Figure 5.12 and Figure 5.13 show respectively the boxplots with the results of the prediction of assembly and welding times. The boxplots are drawn for each implemented prediction model for all projects evaluated in scenario 1.

In line with the **MAPE** for the different experiments shown in table 5.4 the distribution of absolute relative residuals for the assembly time varies between the linear and nonlinear prediction models.

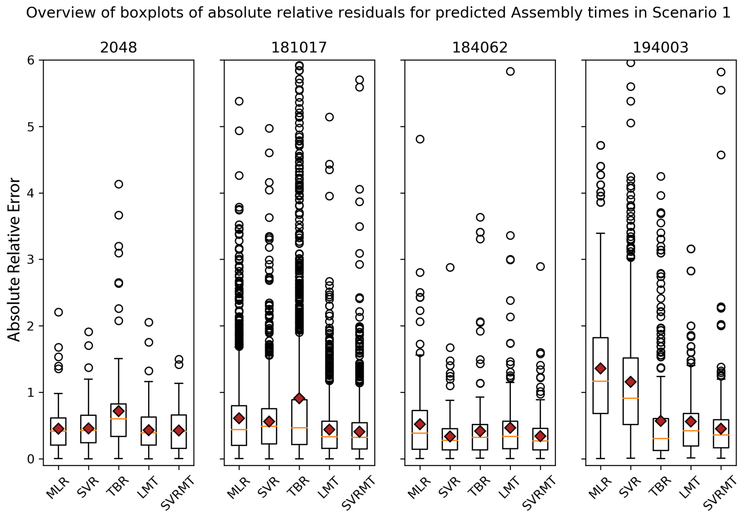


Figure 5.12: Boxplots of the absolute relative residuals for scenario 1 under the implementation of the different assembly time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals.

Table 5.5 shows the results of the proposed **SVRMT** prediction model in scenario 1. For a complete overview of the results for all prediction models in scenario 1, the reader is referred to Appendix C.2.1. It is notable that the standard deviation of the absolute relative residuals for the prediction of assembly times is significantly higher than the residuals of the predicted welding times (respectively 0.40 and 0.30).

This difference can be explained by the nature of the manufacturing steps. As discussed in section 2.3, the assembly step is significantly more influenced by the variations in the products. Additionally, the assembly step is more exposed to errors originated earlier in the manufacturing process. For example, an assembly employee has to assemble multiple parts into one product. If one part is defect / missing, this leads to a disruption in the assembly process, while a welding employee gets an assembled product assigned and focuses on the placement of the final welds.

Furthermore, the R^2 is relatively high for the prediction of both assembly and welding time of the various projects. This implies that the prediction model is able to explain a significant part of the variance in the data. The exception is the R^2 for both the prediction of assembly and welding time for project 194003. This can be explained by the fact that this project has a significant number of repetitive products resulting in equal predicted manufacturing times. There is, however, still a significant number of human related uncertainty in the manufacturing times. In combination with the slightly higher **MAPE** for this project, due to the physical properties varying from the other evaluated projects, this results in a significant lower R^2.

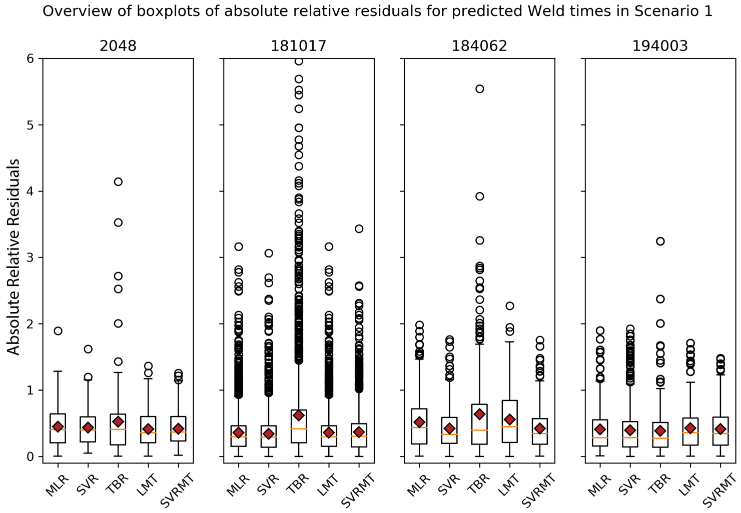


Figure 5.13: Boxplots of the absolute relative residuals for scenario 1 under the implementation of the different welding time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals

Table 5.5: Results for scenario 1 of the SVRMT prediction model for the prediction of the assembly (left) and welding (right) times of the different construction projects

	MAPE	MeAPE	Std	R^2
2048	0.43	0.41	0.30	0.90
181017	0.41	0.32	0.47	0.77
184062	0.34	0.27	0.32	0.92
194003	0.45	0.36	0.49	0.51

	MAPE	MeAPE	Std	R^2
2048	0.42	0.37	0.26	0.84
181017	0.36	0.30	0.32	0.95
184062	0.42	0.35	0.33	0.88
194003	0.41	0.34	0.31	0.46

Comparison to current prediction approach

As stated in section 2.5, it is common for the construction industry to predict manufacturing times for a group of products. Since the proposed prediction model predicts manufacturing time per product, comparing both prediction strategies is not straightforward. In section 2.5, it was also stated that Oostingh Staalbouw predicted manufacturing times per product in 2017 for experimental purposes. Resulting from this experiment, an average **MAPE** of 0.60 was derived for both assembly and welding.

Since project 2048 has been manufactured in 2017, the (experience based) predicted manufacturing times for this project are on product base. An opportunity for comparing the current and proposed approach arises. Table 5.6 shows the results of the different prediction models, along with the current prediction approach for project 2048. For both the prediction of manufacturing times for assembly and welding, the current approach is outperformed by the prediction models in terms of prediction accuracy.

Table 5.6: Results for scenario 1 of the different prediction model for the prediction of the assembly (left) and welding (right) times of project 2048

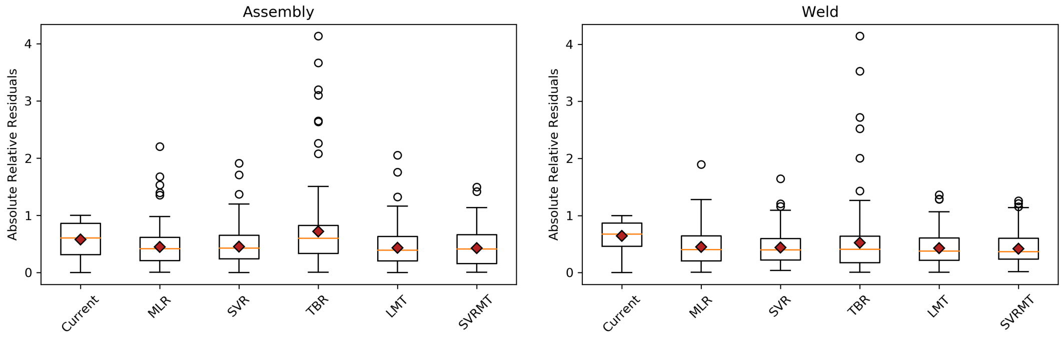
	MAPE	MeAPE	Std	R^2
Current	0.58	0.61	0.31	0.91
MLR	0.45	0.41	0.33	0.92
SVR	0.45	0.43	0.31	0.92
TBR	0.72	0.60	0.69	0.74
LMT	0.43	0.39	0.31	0.92
SVRMT	0.43	0.41	0.30	0.93

	MAPE	MeAPE	Std	R^2
Current	0.64	0.67	0.28	0.80
MLR	0.45	0.40	0.31	0.85
SVR	0.44	0.40	0.28	0.86
TBR	0.53	0.41	0.61	0.67
LMT	0.41	0.36	0.27	0.88
SVRMT	0.42	0.37	0.26	0.88

The boxplots of the absolute relative residuals of the different prediction models are shown in Figure 5.14. Interesting is that for both assembly and welding, the box of the boxplot for the current approach lies significantly higher than the different prediction models. The size of both the box and the whiskers, however, does not differ significantly compared to the tested prediction models. This implies that the absolute relative residuals of the current prediction approach are higher in general, but the distribution of absolute relative residuals around the Mean Absolute Percentage Error is (more or less) the same for all prediction models. Therefore, the different prediction models show an improvement on average prediction accuracy, but not in prediction uncertainty.

Due to unavailable data of predictions from the current approach on a product basis for projects 181017, 184062 and 194003, the assumption that the **MAPE** of 0.60 is representative for new projects is used for comparison.

All in all, the prediction of an entire new project based on historical data from **BIM** and the manufacturing times of other projects results in more accurate predictions. The **MAPE** of the evaluated prediction models is significantly lower than the current, experience based method (0.41/0.60). Overall, the proposed **SVRMT** prediction model yields both most constant and accurate results of the evaluated prediction models. However, using the objective **MAPE** of 0.30, there is still room for improvement.



(a) (b)

Figure 5.14: Boxplot of Absolute Relative Residuals for predicted a) assembly and b) welding times for the evaluated prediction models. Scenario 1, project 2048

5.4.3. Scenario 2 - Per project

Even though the conducted experiments in scenario 1 yielded an improvement over the current experience based prediction method, the results are still off from the set objective **MAPE** of 0.30. In the second scenario, a later stage in the manufacturing phase of the construction projects is regarded. Additionally, the training data for the respective prediction models consists solely of historical data from the project being predicted. For convenience, this experiment is conducted using the 70/30 ratio proposed in the introduction of this section. This corresponds to the situation where 70 % of the project has been manufactured. Analogue to scenario 1, the assembly and welding times are predicted for the products of each project.

The expectation is that the predictions derived in this scenario are more accurate than in scenario 1, since the training data used for the prediction models is expected to be more relevant than data from other projects.

Table 5.7 shows the resulting **MAPE** of the different prediction models for respectively the a) assembly and b) welding steps of the evaluated projects. Additionally, boxplots of the absolute relative residuals for the implemented prediction models of the reviewed projects is provided in Figure 5.15 and Figure 5.16. Prominently project 2048 and project 194003 show significantly different results between scenario 1 and scenario 2. The **MAPE** of project 194003 yields equal results as the indication of best possible accuracy determined in section 2.3.2. This result can be explained by the high ratio of repetitive products occurring in this project.

Project 2048, on the other hand yields significantly worse results than scenario 1. This can possibly be explained by the fact that it is both a relatively small project (small training set) and has a large ratio of unique products along with a wide distribution of both physical properties and manufacturing times.

Project 181017 and project 184062 show similar results with scenario 1.

Table 5.7: MAPE of the results for the different implemented prediction models in scenario 2 for predicting the assembly (left) and welding (right) times. The bold numbers represent the lowest MAPE of the evaluated prediction models per project.

	2048	181017	184062	194003	Average
MLR	0.68	0.46	0.39	0.37	0.48
SVR	0.70	0.45	0.38	**0.24**	0.44
TBR	0.61	0.43	0.54	0.37	0.49
LMT	0.56	**0.39**	0.39	0.31	0.41
SVRMT	**0.52**	**0.39**	**0.36**	**0.24**	**0.38**

	2048	181017	184062	194003	Average
MLR	0.56	0.40	0.39	0.31	0.42
SVR	0.70	0.40	**0.35**	0.29	0.44
TBR	0.63	0.41	0.54	0.32	0.48
LMT	0.56	0.39	0.40	0.28	0.41
SVRMT	**0.51**	**0.37**	0.35	**0.26**	**0.37**

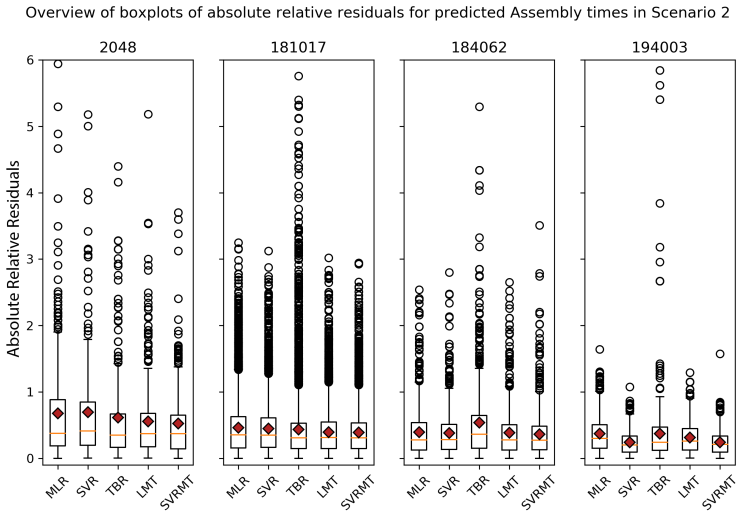


Figure 5.15: Boxplots of the absolute relative residuals for scenario 2 under the implementation of the different assembly time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals

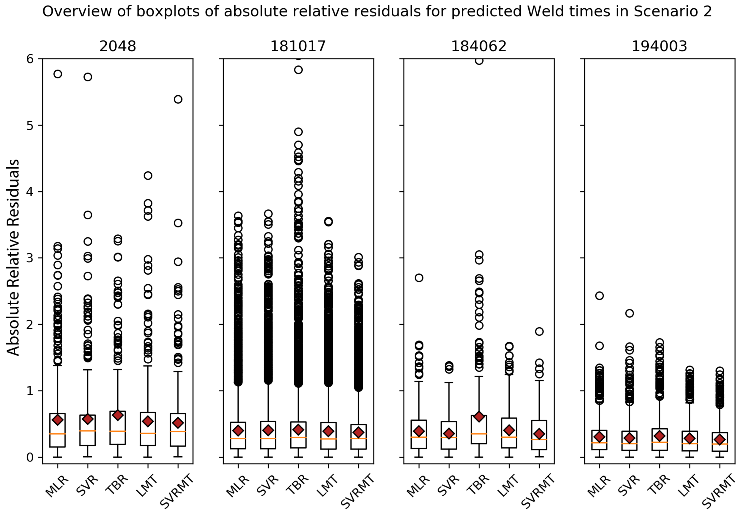


Figure 5.16: Boxplots of the absolute relative residuals for scenario 2 under the implementation of the different welding time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals

5.4.4. Scenario 3 - Mixed projects

Lastly, the available information from scenario 1 and scenario 2 is combined for scenario 3. This results in a training set containing data from prior projects, along with data of realized products of the predicted project. Similar to scenario 2, a 70/30 ratio for the realized products of the predicted project is used, corresponding to

70% of the products already being manufactured.

Table 5.8 shows the **MAPE** of the results for the various prediction models considered in scenario 3. It shows that outlying results of scenario 1 and scenario 2 are straightened out; With the exception of project 194003, the results of scenario 1 and 3 seem to be comparable.

Project 194003 is negatively influenced by data from the other projects, while project 2048 benefits greatly from historical data of the other projects. This is in line with the results of scenario 1 and 2.

Table 5.8: MAPE of the results for the different implemented prediction models in scenario 3 for predicting the assembly (left) and welding (right) times

	2048	**181017**	**184062**	**194003**	**Average**
MLR	0.47	0.51	0.45	0.69	0.53
SVR	0.43	0.42	**0.36**	0.49	0.43
TBR	0.47	0.45	0.48	**0.38**	0.45
LMT	0.41	0.42	0.44	0.41	0.42
SVRMT	**0.40**	**0.36**	0.38	**0.38**	**0.38**

	2048	**181017**	**184062**	**194003**	**Average**
MLR	0.45	0.38	0.42	0.39	0.41
SVR	0.44	0.37	**0.38**	0.34	**0.38**
TBR	0.57	0.41	0.49	**0.27**	0.44
LMT	**0.36**	0.36	0.43	0.36	**0.38**
SVRMT	0.40	**0.36**	0.41	0.34	**0.38**

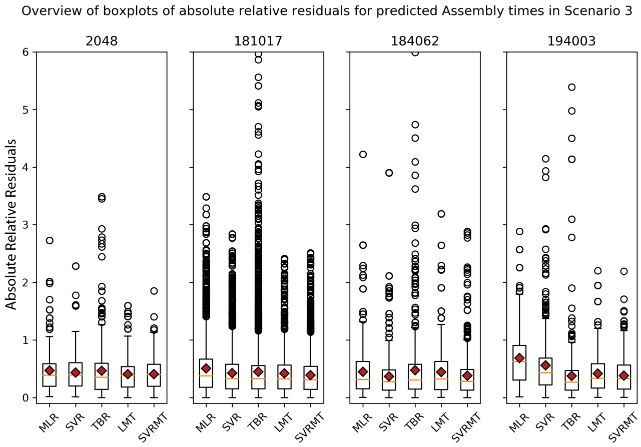


Figure 5.17: Boxplots of the absolute relative residuals for scenario 3 under the implementation of the different assembly time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals

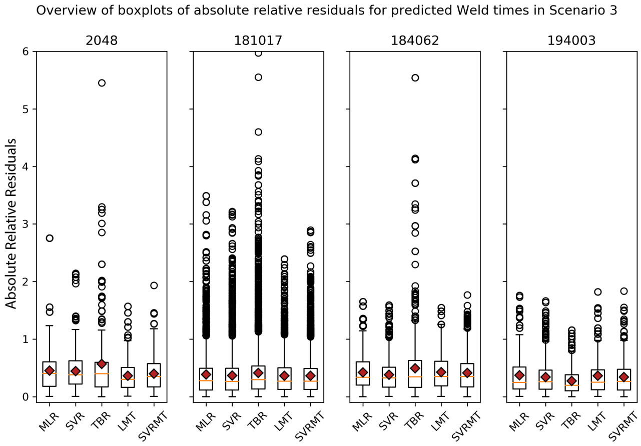
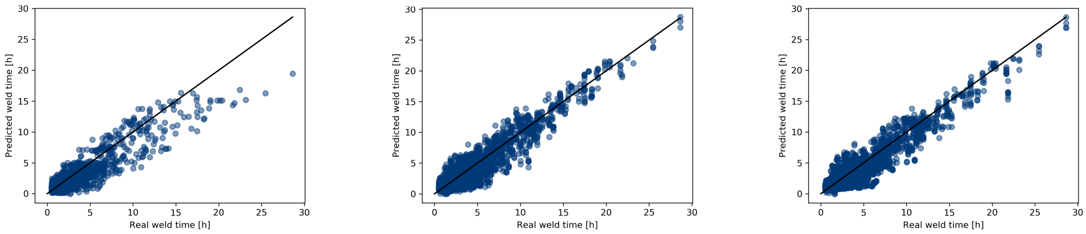


Figure 5.18: Boxplots of the absolute relative residuals for scenario 3 under the implementation of the different welding time prediction models. The diamond shape and the orange line represent respectively the MAPE and MeAPE of the absolute relative residuals

5.4.5. Additional remarks

In addition to the results of the different experimental scenarios described in the above sections, additional findings are encountered. Figure 5.3 shows the inverse fitted value plots for the predicted welding times of the **SVRMT** prediction model for project 181017. This Figure shows the real welding times versus the predicted welding times. Notable is that even though the **MAPE** does not differ significantly between the scenarios, the distribution of the residuals does differ. In scenario 1, the predicted welding times is consistently lower than the realized welding times for realized welding times above 10 hours. This can be explained by the fact that products of project 181017 have on average longer welds than the other evaluated projects. After data of realized products from project 181017 is added to the training data (Figure 5.3 b) and Figure 5.3 c)) the distribution centers more around the 45 degrees line.

Various patterns can be identified for all projects. Therefore, adding more historical data to the training set is recommended. It is expected that increasing the training set will decrease the variance of the residuals. This increase of training data is less likely to reduce the **MAPE** of the predictions, which is more likely to benefit from adding input variables [49].



(a) (b) (c)

Figure 5.19: Inverse fitted value plot for the predicted welding times for project 181017 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model

Furthermore, the Coefficient of Variance (C_v, section 2.4) of the absolute relative residuals of the **SVRMT** prediction model is approximately unity. This is in line with the the C_v derived for the uncertainty in the data (section 2.4). Based on the C_v and the **MeAPE** derived throughout the various scenarios, it can be seen that the **MAPE** is negatively influenced by significant outliers in the data.

Figure 5.20 shows the **SVRMT** prediction model constructed for the prediction of assembly times for project 194003, scenario 1. Notable is that no splits are based on the categorical input variables (profile types). This implies that the influence of profile type on the assembly time is negligible, which is in contrast to the expectation as discussed in section 2.2.3.

In addition, the depth of the **SVRMT** remains relatively low, resulting in a small, interpretable tree. Considering the prerequisite that the prediction model should be easily interpretable, Figure 5.20 ratifies the assumption that **SVRMT** prediction models meet this requirement.

The corresponding coefficients of the **SVR** prediction models of the respective nodes are provided in table 5.9. Notable is that the no. holes is dropped in each prediction model, which is in line with the relatively low correlation coefficient between physical properties and manufacturing time, derived in section 2.4. Furthermore, the coefficients differ between the respective nodes, showing the nonlinearity of the relationship between the physical properties and assembly time.

Table 5.9: Coefficients for the SVRMT model used for the predictions of assembly times for project 194003, scenario 1

	Weight [kg]	**Length [mm]**	**Weld length [mm]**	**No. Parts**	**No. Welds**	**No. Holes**
Root	2,94E-04	3,67E-05	4,18E-05	8,46E-02	9,64E-03	
Node 1		1,58E-05	2,38E-05	1,66E-02	8,43E-03	
Node 2	4,75E-04	3,54E-05	3,54E-05	8,08E-02	9,13E-03	
Node 3	9,55E-04		4,15E-05		1,13E-02	
Node 4	-1,62E-04	2,50E-04	3,74E-05	8,41E-02	2,52E-03	
Node 5	4,95E-05	1,77E-04	1,63E-05	2,31E-01	-7,75E-02	
Node 6	6,55E-04	5,90E-04	4,19E-04	6,06E-02	2,15E-02	

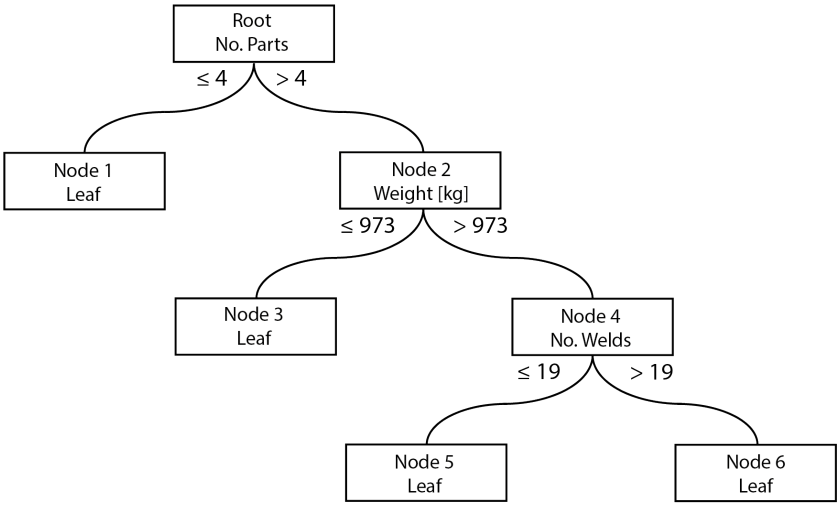


Figure 5.20: Example of SVRMT for the prediction of assembly times in scenario 1

Figure 5.21 shows the **SVRMT** model constructed for the prediction of welding times of project 194003, scenario 1. Notable is that this tree is smaller than the **SVRMT** for the prediction of assembly times. From the results shown in table 5.4, the **SVR** is slightly more accurate than the **SVRMT** prediction model (for project 194003 in scenario 1). This result, along with the tree shown in Figure 5.21, shows that small overfitting occurred during the construction of the model. The coefficients of the **SVR** prediction models differ only slightly across the different nodes of the tree. In combination with the small difference in **MAPE** between the **SVR** and **SVRMT** prediction models, it can be seen that the effect of overfitting is limited.

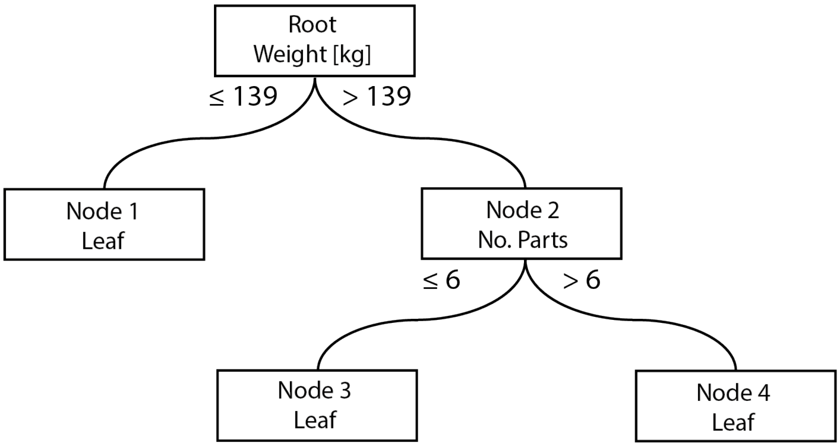


Figure 5.21: Example of SVRMT for the prediction of welding times in scenario 1

Table 5.10: Coefficients for the SVRMT model used for the predictions of welding times for project 194003, scenario 1

	Weight [kg]	WeldLength [mm]	No. Parts	Length [mm]	No. Welds	No. Holes
Root	4,81E-04	2,03E-04	3,37E-02	-2,91E-05	1,64E-02	
Node 1	3,50E-03	4,87E-05	3,37E-02	-8,30E-05	-1,52E-03	
Node 2	3,96E-04	2,09E-04	2,91E-02	-1,20E-05	1,85E-02	
Node 3	9,14E-05	1,88E-04	4,35E-02	4,17E-07	4,26E-02	
Node 4	9,42E-04	2,01E-04	1,88E-02	-3,25E-05	4,55E-03	

5.5. Conclusion

In this chapter, the sub-question: *"How can the conceptual prediction model be validated?"* is discussed. The conceptual prediction model is validated through two types of experiments. At first, the proposed prediction model is verified in hypothetical situations. Afterwards the proposed prediction model is tested in several real case scenarios.

Prior to the conduction of the validation experiments, the sub-question: *"How can the performance of prediction models be evaluated and compared?"* is regarded. In order to evaluate different prediction models, several methods have been identified. A commonly used Performance Indicator is the Coefficient of determination (R^2), which expresses how much variance in the data is explained by the prediction model. Furthermore, the Mean Absolute Percentage Error (**MAPE**) is used to express the average relative error of the model. The **MAPE** is especially useful to evaluate the performance of prediction models in different data sets.

Besides metrical Performance Indicators, visual methods can be used to evaluate the performance of a prediction model. Plotting the real value against the predicted value can be used to evaluate the ability of the prediction model to reconstruct the real relationship. In addition, boxplots of residuals can be used to visually compare the distribution of residuals from different prediction models.

After relevant Performance Indicators have been identified, the proposed Support Vector Regression Model Tree (**SVRMT**) is tested in several hypothetical situations. During this experimental phase, the assumptions that the **SVRMT** prediction model is able to yield accurate predictions for both linear and nonlinear relationships with significant added noise are verified.

Upon the verification of the proposed **SVRMT** prediction model in hypothetical situations, the prediction model is tested in three real case scenarios. These scenarios correspond to 1) the start of a new construction project, 2) 70 % of the products for a construction project manufactured and 3) a scenario where information from both historical and 70% of the project to be predicted is available. Along with the **SVRMT** prediction model, Multiple Linear Regression (**MLR**), Support Vector Regression (**SVR**), Tree Based Regression (**TBR**) and Linear Model Tree (**LMT**) prediction models are tested for comparison.

For scenario 1, predictions of manufacturing times based on historical data from construction products have increased accuracy compared to the currently used, experience based approach. However, it is notable that the prediction accuracy decreases if the physical properties of the project differ significantly from historical projects.

In scenario 2, increased accuracy compared to scenario 1 is yielded for projects with a high ratio of repetitive products. For relatively small projects with a small ratio of similar products, the accuracy decreases significantly.

Overall, scenario 3 did not yield increased prediction accuracy compared to scenario 1 and scenario 2. However, outliers (both positive and negative) are flattened out in this scenario, leading to most consistent predictions in terms of accuracy.

Based on these results. it would be interesting to extend the database with projects from the past in order to increase the variations in training data for the prediction model. It is expected that this increase of training data leads to more consistent predictions (smaller distribution of residuals) for all types of construction projects. In addition, it is expected that extending the input variables will increase the average prediction accuracy.

In addition, it would be interesting to investigate scenario 2 and 3 in more detail by conducting experiments, for example after 25% or 50% of the products being manufactured. This way, more insight in the increasing accuracy with increasing part of the project manufactured can be obtained.

In general, in each scenario, most tested prediction models yield more accurate predictions than the current, experience based manufacturing time predictions (**MAPE** of 0.60). Overall, the proposed **SVRMT** prediction model yielded both slightly more accurate and constant results predictions compared to the other evaluated prediction models. With a yielded **MAPE** of 0.41, 0.38 and 0.38 for respectively scenario 1, scenario 2 and scenario 3, the **SVRMT** prediction model is a significant improvement compared to the current prediction approach.

Even though there is still room for improvement (objective **MAPE** of 0.30 not met), it can be concluded that it would be beneficial to implement the proposed **SVRMT** prediction model to improve the prediction accuracy for manufacturing times per manufacturing step.

6 Conclusion

Currently, the construction industry predicts manufacturing times of structural elements (products) based on the experience of shop managers. This approach is prone to errors, leading to ineffective manufacturing schedules. The company used for this case study, Oostingh Staalbouw, currently yields a Mean Absolute Percentage Error (**MAPE**) of 0.60 (60%) for the prediction of manufacturing times per manufacturing step.

The past decade, the construction industry started using Building Information Models (**BIM**) to centralize storage of information about construction projects. Along with this development, the application of data analysis in the manufacturing industry grew rapidly.

The aim of this research has been to combine both developments by proposing a general manufacturing time prediction model based on information stored in **BIM** along with manufacturing data. For the case study, the objective was to reduce the **MAPE** of the predictions from 0.60 to 0.30. In order to reach this objective, the following research question is formulated:

"How to develop a manufacturing time prediction model, using ***BIM*** *and manufacturing data, in order to create more effective manufacturing schedules?"*

Even though a general manufacturing time prediction model is to be proposed, the scope of this research has been limited to manufacturing steps with high human involvement. The proposed manufacturing time prediction model is validated in a case study in collaboration with Oostingh Staalbouw. Since the proceedings of Oostingh Staalbouw are limited to the steel elements of constructions, validation of the prediction model is only performed on the manufacturing of steel elements (products). It is however assumed that the proposed prediction model is suitable for the manufacturing of all aspects of the construction industry.

The road towards answering the research question can be found by answering several sub questions:

What does the manufacturing process at Oostingh Staalbouw look like?
The manufacturing process of Oostingh Staalbouw can be divided in four steps. In the preprocessing step, standard profiles are customized for the product. Afterwards, during the assembly step, the separate parts for the products are assembled. In the welding step, the final welds are placed on the product. Finally, products can be coated in the coating step. Since this research focuses on manufacturing steps with high human involvement, combined with the lack of accurate data from the coating step, the remainder of this research focused on the prediction of manufacturing times for the assembly and welding step.

Which data can be extracted from ***BIM*** *and the manufacturing process?*
Even though **BIM** is capable of storing any kind of information, the actual stored information is currently still limited to physical properties of products. Both quantitative and categorical properties can be identified.

Using Spearman's correlation coefficient, monotonic relationships can be identified between different quantitative physical properties and manufacturing times. The drawback of this approach is that it remains ambiguous whether this relationship is linear or nonlinear.

The identified categorical properties are the different profile types; H-beams, U-beams, Hollow sections and Plates. It is expected that the handling of these profiles differs, having an impact on the manufacturing time of the product.

In addition, data from the manufacturing process consists of manufacturing times. These manufacturing times are measured by scanning barcodes coupled to the product, at the start and at the end of the manufacturing step. After each work shift, the scanned data is synchronized with BIM. Repetitive manufactured products are analyzed in order to get insight in the size of human related uncertainty in the data. The manufacturing time of similar products are predicted using the mean manufacturing time of these repetitive products. The Mean Absolute Percentage Error (**MAPE**) of this approach turned out to be 0.25 for both the assembly and welding step. This approach, however, is unsuitable for the prediction of manufacturing times of completely new, unique products. Therefore, it is expected that the **MAPE** of the predictions will be higher than 0.25, leading to the objective **MAPE** of 0.30.

Based on the characteristics of the available data, several prerequisites for the prediction model are derived. The prediction model should be able to deal with quantitative and categorical input variables. Additionally, the prediction model should be able to predict both linear and nonlinear relationships between the input variables and manufacturing times. Furthermore, due to uncertainty in the available data, the prediction model should be robust against this uncertainty.

In order to increase the rate of acceptance of the prediction model, the prediction model should be easily interpretable.

Which manufacturing time prediction models are available in literature?
After determining the prerequisites for the prediction model, related research on the prediction of manufacturing times is reviewed. Research on using **BIM** to predict manufacturing times is limited to one study by Hu et al. [26]. In this study, the implementation of a Multiple Linear Regression (**MLR**) model resulted in significantly improved prediction accuracy compared to the experience based approach currently used in the construction industry. The accuracy of this approach, however, is strongly depending on the linearity of the relationship between physical properties and manufacturing times.

In comparable Engineered-to-Order industries (**ETO**), prediction models like Support Vector Regression (**SVR**), Tree Based Regression (**TBR**) and Neural Networks (**NN**) are used for the prediction of product lead time. Even though the product lead time is different from the manufacturing time per manufacturing step, it would be interesting to evaluate these prediction models for the prediction of manufacturing times per manufacturing step. Additionally, a prediction model named Linear Model Tree (**LMT**) is found in literature, which combines advantages of **MLR** and **TBR**.

Based on the prerequisites for the prediction model and the available prediction models identified in the literature, the **LMT** prediction model is chosen. Like **TBR**, the data is split in a decision tree like structure. However, the **LMT** prediction model uses linear models in its nodes rather than the mean value of data reaching the node. This way, the **LMT** prediction model is able to predict continuous values rather than discrete values, making it more applicable for predicting manufacturing times of new products.

The drawback of the **LMT** is that it uses **MLR** prediction models in its nodes, which are sensitive to uncertainty in the data. In this research, an adaptation is proposed which combines advantages of **SVR** and **LMT** prediction models. **SVR** prediction models tend to outperform **MLR** prediction models if uncertainty in the data arise. Therefore, the **MLR** prediction models in the nodes are replaced by **SVR** prediction models. It was therefore expected that the proposed prediction model (Support Vector Regression Model Tree (**SVRMT**)) is more robust against uncertainty in the data compared to the **LMT** prediction model.

Which conceptual prediction model can best be used to predict manufacturing times?
For the construction of a **SVRMT** prediction model several steps can be identified. At first, nonlinear data is split into linear segments. Afterwards, an **SVR** prediction model is placed in each node of the model tree. At last, the **SVRMT** is pruned in order to prevent overfitting.

For the prediction of new data, the model tree is descended to the corresponding leaf. After the leaf has been reached, the manufacturing time is predicted using the corresponding **SVR** prediction model. In order to compensate for sharp discontinuities between adjacent leaves in the model tree, the predictions are smoothed. In this smoothing the model tree is ascended from leaf to root. At each intermediate node, the prediction is adjusted using the **SVR** prediction model of the node. After reaching the root of the tree, the resulting prediction is the output of the prediction model.

How can the performance of prediction models be evaluated and compared?
For the evaluation and comparison of prediction models several methods can be used. The coefficient of determination (R^2) might be the most widely used metric for this purpose. This Performance Indicator, however, might imply that the prediction model fits the relationship well, even though the prediction model is overfitted to uncertainty in the data.

The Mean Absolute Percentage Error (**MAPE**) can be used to compare results of prediction models across different datasets. Since this metric shows the relative error, it provides insight in the size of the prediction error. This Performance Indicator focuses on the reduction of the prediction error and is therefore identified as main Performance Indicator of this study.

In addition, visual approaches for the evaluation of different prediction models can be identified. The ability of the prediction model to accurately reconstruct a relationship between input and output can be visualized using inverse fitted value plots. The comparison of the distribution of residuals from the implementation of different prediction models can be done by using boxplots.

How can the conceptual prediction model be validated?
For the validation of the **SVRMT** prediction model two experimental phases are performed.

At first, the assumptions that the proposed **SVRMT** prediction model is able to deal with both linear and nonlinear relationships under the influence of uncertainty in the data is verified. During these verification experiments, a linear, quadratic and step relationship with added heteroskedastic noise are reconstructed by the proposed **SVRMT** prediction model. This prediction model is compared to the prediction models identified in related literature. The conducted experiments indicate that the **SVRMT** yields the most accurate reproductions of the evaluated prediction models.

After this verification, the proposed prediction model is validated using data from Oostingh Staalbouw. For this validation, four construction projects from the recent past (September 2018 - June 2019) are tested. Three scenarios have been considered; the first scenario corresponds to the kick-off of the manufacturing of a new project. At this stage, the available data is limited to historical data from other projects. This data is used to train the prediction model. Afterwards, both assembly and welding times are predicted for the new project. This experiment is repeated for all projects being left out once. The results for scenario 1 showed a significant improvement over the currently used experience based approach. The **MAPE** of the predictions for this experiments yielded an average accuracy of 0.41 (compared to 0.60 currently).

In the second scenario, a progressed stage of the manufacturing of a project is considered. For this scenario, it is assumed that 70% of the project has already been completed. The prediction model is trained using the data of realized products of the project. This turned out to be especially useful for projects differing significantly from other projects. In addition, projects with a significant number of repetitive products yielded increased prediction accuracy compared to scenario 1. Relatively small projects with limited repetition, on the other hand, yielded decreased prediction accuracy compared to scenario 1.

Finally, a combination of scenario 1 and scenario 2 has been investigated. For this scenario, data from historical projects, along with realized data of the project to be predicted is used to predict the manufacturing times of the remaining products. Overall, outlying results (both positive and negative) are flattened out. The average **MAPE** increased slightly compared to scenario 1 and scenario 2. Most notable is the increased consistency of prediction accuracy across the different projects evaluated in scenario 3.

Overall, all evaluated prediction models resulted in increased accuracy for the prediction of manufacturing times per manufacturing step compared to the current, experience based approach. Of all prediction models, the proposed **SVRMT** prediction model showed both most constant and slightly more accurate results. Even though the **SVRMT** prediction model yielded significantly more accurate predictions, the MAPE is still higher than the objective.

Through this research, it can be concluded that there is significant room for improvement in the accuracy of predicting manufacturing times. Compared to the current, experience based approach taken in the construction industry, the implementation of prediction models based on historical data from both **BIM** and the manufacturing process yielded a significant improvement in terms of prediction accuracy. During this research, the **MAPE** of predictions has been reduced from 0.60 to 0.38. This is a significant improvement compared to the **MAPE** of the currently used approach. The proposed **SVRMT** showed in both hypothetical and real-case scenarios to be able to yield most accurate predictions of the reviewed prediction models.

Recommendations

Even though the results of this research showed a significant improvement compared to the current approach, the objective **MAPE** of 0.30 has not been met. Based on the findings of this research, several recommendations for further research can be proposed.

At first, the effect of increased accuracy in predicted manufacturing times on manufacturing schedules should be studied. It was assumed that increased accuracy in predicted manufacturing times would result in more effective manufacturing schedules. It is therefore recommended to verify this assumption by conducting further research on the effect of more accurate predicted manufacturing times on manufacturing schedules.

Furthermore, it would be interesting to test whether the proposed prediction model yields an increase of prediction accuracy for other manufacturing processes in the construction industry. This way, the assumption that the proposed manufacturing time prediction model is indeed a general model can be verified.

The implementation of different heuristics used for the construction of the **SVRMT** can be further studied. In this research, a greedy, top down approach was used to construct a **SVRMT**. Heuristics based on evolutionary algorithms has been proposed in a series of papers by Kretowski and Czajkowski [35], Czajkowski and Kretowski [13] and Czajkowski and Kretowski [14]. This might especially be interesting in case of a significantly large set of historical data under computational time constraints.

In this research, the input variables are limited to information currently stored in **BIM**. This information encompasses the physical properties of products. It would be interesting to relax the assumption that manufacturing times of manufacturing steps are not influenced by process or external variables. Input variables, like day of the week, weather conditions, the number of products scheduled for the day etc. can be taken into account to study the influence of these variables on the accuracy of the prediction model.

Additional insight in the applicability of the prediction model, scenario 2 and scenario 3 can be evaluated further. Especially for projects differing significantly from historical projects, it would be interesting to gain insight in the turning point for using only information of the regarded project.

From the same perspective, scenario 3 can be investigated further. In that case it would be interesting to study the effect of a weighing function based on the relationship to the products to be predicted. Products of the same project can be given more emphasis in the prediction model than products from historical projects.

Last but not least, the predictions of manufacturing times per manufacturing step based on the physical properties of products can be used to optimize the design of construction projects in terms of manufacturing costs. During the design phase, the prediction model can predict the manufacturing time of the conceptual design. In combination with insight of the effect of the various physical properties on the manufacturing times, more sophisticated considerations on the design can be made.

Bibliography

[1] Hervé Abdi. Coefficient of variation. *Encyclopedia of research design*, 1:169–171, 2010. doi: http://dx.doi.org/10.4135/9781412961288.n56.

[2] Federico Adrodegari, Andrea Bacchetti, Roberto Pinto, Fabiana Pirola, and Massimo Zanardini. Engineer-to-order (ETO) production planning and control: an empirical framework for machinery-building companies. *Prod. Plan. Control*, 26(11):910–932, aug 2015. ISSN 0953-7287. doi: 10.1080/09537287.2014.1001808. URL `https://doi.org/10.1080/09537287.2014.1001808`.

[3] Anaconda. Anaconda Software Distribution, 2016. URL `https://anaconda.com`. Accessed on: 2019-10-10.

[4] Mohsenijam Arash, Siu Ming-Fung Francis, and Lu Ming. Modified Stepwise Regression Approach to Streamlining Predictive Analytics for Construction Engineering Applications. *J. Comput. Civ. Eng.*, 31(3): 04016066, may 2017. doi: 10.1061/(ASCE)CP.1943-5487.0000636. URL `https://doi.org/10.1061/(ASCE)CP.1943-5487.0000636`.

[5] Mariette Awad and Rahul Khanna. Support Vector Regression. In *Effic. Learn. Mach. Theor. Concepts, Appl. Eng. Syst. Des.*, pages 67–80. Apress, Berkeley, CA, 2015. ISBN 978-1-4302-5990-9. doi: 10.1007/978-1-4302-5990-9_4. URL `https://doi.org/10.1007/978-1-4302-5990-9{_}4`.

[6] B2Bmetal. HEB beams, European standard wide flange H beams, dimensions, specifications. HE B beams in accordance with former standard Euronorm 53-62, 2019. URL `http://www.b2bmetal.eu/heb-sections-specification`. Accessed on: 2019-08-12.

[7] B2Bmetal. Square Hollow Structural Sections - HSS, EN 10219:1997 Cold Formed steel square sections, 2019. URL `http://www.b2bmetal.eu/en/pages/index/index/id/65/`. Accessed on: 2019-08-12.

[8] B2Bmetal. UPE European standard U channels (U profile) with parallel flanges. UPE steel beam specifications, dimensions, properties, 2019. URL `http://www.b2bmetal.eu/u-sections-upe-specification`. Accessed on: 2019-08-12.

[9] F T S Chan. Performance Measurement in a Supply Chain. *Int. J. Adv. Manuf. Technol.*, 21(7):534–548, may 2003. ISSN 1433-3015. doi: 10.1007/s001700300063. URL `https://doi.org/10.1007/s001700300063`.

[10] Peter Chapman, Janet M. Clinton, Randy Kerber, Tom Khabaza, Thomas Reinartz, Christopher R. Shearer, and Richard Wirth. CRISP-DM 1.0: Step-by-step data mining guide. 2000.

[11] Ying Cheng, Ken Chen, Hemeng Sun, Yongping Zhang, and Fei Tao. Data and knowledge mining with big data towards smart production. *J. Ind. Inf. Integr.*, 9:1–13, mar 2018. ISSN 2452-414X. doi: 10.1016/J.JII.2017.08.001. URL `https://www.sciencedirect.com/science/article/pii/S2452414X17300584{#}fig0002`.

[12] Vladimir Cherkassky and Yunqian Ma. Practical selection of SVM parameters and noise estimation for SVM regression. *Neural Networks*, 17(1):113–126, jan 2004. ISSN 0893-6080. doi: 10.1016/S0893-6080(03)00169-2. URL `https://www.sciencedirect.com/science/article/pii/S0893608003001692?via{%}3Dihub`.

[13] Marcin Czajkowski and Marek Kretowski. An Evolutionary Algorithm for Global Induction of Regression Trees with Multivariate Linear Models. In Kryszkiewicz Marzena, , Henryk Rybinski, and Skowron Andrzej, and and Raś Zbigniew W, editors, *Found. Intell. Syst.*, pages 230–239, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. ISBN 978-3-642-21916-0.

[14] Marcin Czajkowski and Marek Kretowski. Evolutionary induction of global model trees with specialized operators and memetic extensions. *Inf. Sci. (Ny).*, 2014. ISSN 00200255. doi: 10.1016/j.ins.2014.07.051.

[15] F.M. Dekking, C. Kraaikamp, H.P. Lopuhaä, and L.E. Meester. *A Modern Introduction to Probability and Statistics: understanding why and how.* Springer, London, 1 edition, 2005. ISBN 1-85233-896-2.

[16] A V Dorugade and D N Kashid. Alternative method for choosing ridge parameter for regression. *Appl. Math. Sci.*, 4(9-12):447–456, 2010. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-77950950640{&}partnerID=40{&}md5=58f540b236e9e26c767d35515a9a64c2.

[17] Harris Drucker, Christopher J C Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Adv. Neural Inf. Process. Syst.*, pages 155–161, 1997.

[18] Chuck Eastman, Paul Teicholz, Rafael Sacks, and Kathleen Liston. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors.* John Wiley & Sons, 2011. ISBN 111802169X.

[19] John Fox. *Regression diagnostics: An introduction*, volume 79. Sage, 1991. ISBN 080393971X.

[20] David A. Freedman. *Statistical Models: Theory and Practice.* Cambridge University Press, Leiden, 2nd edition, 2009. ISBN 9780511604140 0511604149.

[21] Damodar N. Gujarati and Dawn C. Porter. *Basic Econometrics.* McGraw-Hill Irwin, New York, fith edition, 2009. ISBN 978-0-07-337577-9.

[22] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[23] Dávid Gyulai, András Pfeiffer, Gábor Nick, Viola Gallina, and Wilfried Sihn. Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *IFAC-PapersOnLine*, 51 (11):1029–1034, jan 2018. ISSN 2405-8963. doi: 10.1016/J.IFACOL.2018.08.472. URL https://www.sciencedirect.com/science/article/pii/S2405896318316008?via{%}3Dihub.

[24] Thomas Haslwanter. *An Introduction to Statistics with Python - 2016.* Springer, Cham, 2016. ISBN 978-3-319-28315-9. doi: https://doi.org/10.1007/978-3-319-28316-6.

[25] Simon Haykin. *Neural Networks and Learning Machines.* Pearson, New Jersey, third edition, 2008. ISBN 978-0-13-147139-9.

[26] X Hu, M Lu, and S Abourizk. BIM-based data mining approach to estimating job man-hour requirements in structural steel fabrication. In *Proc. - Winter Simul. Conf.*, volume 2015-January, pages 3399–3410, 2015. doi: 10.1109/WSC.2014.7020173. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84940519162{&}doi=10.1109{%}2FWSC.2014.7020173{&}partnerID=40{&}md5=af32c2a66c7e6ba57b594e6a50c93b59.

[27] Minhoe Hur, Seung-kyung Lee, Bongseok Kim, Sungzoon Cho, Dongha Lee, and Daehyung Lee. A study on the man-hour prediction system for shipbuilding. *J. Intell. Manuf.*, 26(6):1267–1279, 2015. ISSN 1572-8145. doi: 10.1007/s10845-013-0858-3. URL https://doi.org/10.1007/s10845-013-0858-3.

[28] Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

[29] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, New York, NY, New York, 2013. ISBN 978-1-4614-7138-7. doi: https://doi.org/10.1007/978-1-4614-7138-7.

[30] Jansen Building Systems. Design: BIM models of Jansen Steel Systems – Jansen AG, 2019. URL https://www.jansen.com/en/building-systems-profile-systems-steel/services-steel-systems/planning-bim-steel-systems.html. Accessed on: 2019-08-13.

[31] Aram Karalič. Employing Linear Regression in Regression Tree Leaves. In *Proc. 10th Eur. Conf. Artif. Intell.*, ECAI '92, pages 440–441, New York, NY, USA, 1992. John Wiley & Sons, Inc. ISBN 0-471-93608-1. URL http://dl.acm.org/citation.cfm?id=145448.146775.

[32] Azme Khamis, Zuhaimy Ismail, Khalid Haron, and Ahmanad Tarmizi Mohammed. The Effects of Outliers Data on Neural Network Performance. *J. Appl. Sci.*, (5):1394–1398, 2005. doi: 10.3923/jas.2005.1394.1398.

[33] Will Koehrsen. Overfitting vs. Underfitting: A Complete Example, 2018. URL `https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765`. Accessed on: 2019-04-25.

[34] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[35] Marek Kretowski and Marcin Czajkowski. An Evolutionary Algorithm for Global Induction of Regression Trees. In Rutkowski Leszek, , Rafał Scherer, and Tadeusiewicz Ryszard, and Zadeh Lotfi A., and and Zurada Jacek M, editors, *Artifical Intell. Soft Comput.*, pages 157–164, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-13232-2.

[36] Junbok Lee, Young-Jin Park, Chang-Hoon Choi, and Choong-Hee Han. BIM-assisted labor productivity measurement method for structural formwork. *Autom. Constr.*, 84:121–132, dec 2017. ISSN 0926-5805. doi: 10.1016/J.AUTCON.2017.08.009. URL `https://www.sciencedirect.com/science/article/pii/S0926580517307264?via{%}3Dihub`.

[37] Jeffrey K Liker. *The Toyota way*. McGraw-Hill, New York, first edition, 2004. ISBN 0-07-139231-9.

[38] Lukas Lingitz, Viola Gallina, Fazel Ansari, Dávid Gyulai, András Pfeiffer, Wilfried Sihn, and László Monostori. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia CIRP*, 72:1051–1056, jan 2018. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2018.03.148. URL `https://www.sciencedirect.com/science/article/pii/S2212827118303056?via{%}3Dihub`.

[39] H Liu, M S Altaf, Z Lei, M Lu, and M Al-Hussein. Automated production planning in panelized construction enabled by integrating discrete-event simulation and BIM. In *5th Int. Constr. Spec. Conf.*, pages 8–10, 2015.

[40] Arash Mohsenijam and Ming Lu. Achieving Sustainable Structural Steel Design by Estimating Fabrication Labor Cost Based on BIM Data. *Procedia Eng.*, 145:654–661, jan 2016. ISSN 1877-7058. doi: 10.1016/J.PROENG.2016.04.056. URL `https://www.sciencedirect.com/science/article/pii/S1877705816300613?via{%}3Dihub`.

[41] M M Mukaka. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi Med. J.*, 24(3):69–71, sep 2012. ISSN 1995-7270. URL `https://www.ncbi.nlm.nih.gov/pubmed/23638278https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/`.

[42] Arnaud de Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean Absolute Percentage Error for regression models. *Neurocomputing*, 192:38–48, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.12.114. URL `http://www.sciencedirect.com/science/article/pii/S0925231216003325`.

[43] Satoshi Nagahara and Youichi Nonaka. Product-specific Process Time Estimation from Incomplete Point of Production Data for Mass Customization. *Procedia CIRP*, 67:558–562, jan 2018. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2017.12.260. URL `https://www.sciencedirect.com/science/article/pii/S2212827117312064?via{%}3Dihub`.

[44] G A Peñaloza, D D Viana, F S Bataglin, C T Formoso, and I R Bulhões. Guidelines for integrated production control in engineer-to-order prefabricated concrete building systems: Preliminary results. In *IGLC 2016 - 24th Annu. Conf. Int. Gr. Lean Constr.*, pages 103–112, 2016. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84995923658{&}partnerID=40{&}md5=6e21df8bac776613704667e858bd8f3e`.

[45] András Pfeiffer, Dávid Gyulai, Botond Kádár, and László Monostori. Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Methods. *Procedia CIRP*, 41:75–80, jan 2016. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2015.12.018. URL `https://www.sciencedirect.com/science/article/pii/S2212827115010975?via{%}3Dihub`.

[46] John R Quinlan. Learning with continuous classes. In *5th Aust. Jt. Conf. Artif. Intell.*, volume 92, pages 343–348. World Scientific, 1992.

[47] Erwin Rauch, Patrick Dallasega, and Dominik T. Matt. Synchronization of Engineering, Manufacturing and on-site Installation in Lean ETO-Enterprises. *Procedia CIRP*, 37:128–133, jan 2015. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2015.08.047. URL `https://www.sciencedirect.com/science/article/pii/S2212827115009002?via{%}3Dihub`.

[48] Thomas P. Ryan. *Modern regression methods.* Wiley, Hoboken, N.J., 2nd edition, 2009. ISBN 978-0-470-08186-0.

[49] Justin D. Salciccioli, Yves Crutain, Matthieu Komorowski, and Dominic C. Marshall. *Sensitivity Analysis and Model Validation*, pages 263–271. Springer International Publishing, Cham, 2016. ISBN 978-3-319-43742-2. doi: 10.1007/978-3-319-43742-2_17. URL `https://doi.org/10.1007/978-3-319-43742-2_17`.

[50] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. ISSN 0893-6080.

[51] Johan A.K. Suykens. Support Vector Machines: A Nonlinear Modelling and Control Perspective. *Eur. J. Control*, 7(2-3):311–327, jan 2001. ISSN 0947-3580. doi: 10.3166/EJC.7.311-327. URL `https://www.sciencedirect.com/science/article/pii/S0947358001711521`.

[52] Tekla. What is BIM (Building Information Management)?, 2019. URL `https://www.tekla.com/about/what-is-bim`. 2019-02-14.

[53] Robert Tibshirani. Regression Shrinkage and Selection Via the Lasso. *J. R. Stat. Soc. Ser. B*, 58(1):267–288, jan 1996. ISSN 0035-9246. doi: 10.1111/j.2517-6161.1996.tb02080.x. URL `https://doi.org/10.1111/j.2517-6161.1996.tb02080.x`.

[54] Israel Tirkel. Forecasting flow time in semiconductor manufacturing using knowledge discovery in databases. *Int. J. Prod. Res.*, 51(18):5536–5548, sep 2013. ISSN 0020-7543. doi: 10.1080/00207543.2013.787168. URL `https://doi.org/10.1080/00207543.2013.787168`.

[55] D D Viana, I R Bulhões, and C T Formoso. Guidelines for integrated planning and control of engineer-to-order prefabrication systems. In *21st Annu. Conf. Int. Gr. Lean Constr. 2013, IGLC 2013*, pages 486–495, 2013. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84903288681{&}partnerID=40{&}md5=c4319ff53209289424e8fb4cb4f98f9d`.

[56] Yong Wang and Ian H Witten. Induction of model trees for predicting continuous classes. Technical report, 1996. URL `https://hdl.handle.net/10289/1183`.

[57] Ansong Wong. Introduction to Model Trees from scratch, 2018. URL `https://towardsdatascience.com/introduction-to-model-trees-6e396259379a`. Accessed on: 2019-05-07.

[58] T Yu and H Cai. The prediction of the man-hour in aircraft assembly based on support vector machine particle swarm optimization. *J. Aerosp. Technol. Manag.*, 7(1):19–30, 2015. doi: 10.5028/jatm.v7i1.409. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924762303{&}doi=10.5028{%}2Fjatm.v7i1.409{&}partnerID=40{&}md5=518371d17a5e838ac5f637da354699e4`.

[59] Hui Zou and Trevor Hastie. Regularization and variable selection via the Elastic Net. *J. R. Stat. Soc. Ser. B*, 67:301–320, 2005.

Appendices

A

Academic Paper

Prediction of Manufacturing Times using Building Information Models

L.A. van der Plas
dr. ir. X. Jiang
prof. dr. R.R. Negenborn

Delft University of Technology
L.A.vanderPlas@student.tudelft.nl

October 15, 2019

Abstract

In the construction industry, it is common to predict manufacturing times of structural elements based on the experience of shop managers. With the increasing use of Building Information Models (BIM) and data analysis in manufacturing processes an opportunity for improving the accuracy of predicted manufacturing times arises. This research proposes a general prediction model based on physical properties of structural elements extracted from BIM, along with manufacturing times of manufactured elements. The proposed Support Vector Regression Model Tree (SVRMT) is tested in both hypothetical and real case scenarios. Through validation in real case scenarios, the SVRMT prediction model turned out to be a significant improvement in terms of prediction accuracy, compared to the current experience based approach.

I. Introduction

The Engineered-To-Order nature of unique complex steel structures requires different structural elements (hereafter referred to as "products") for each construction project. Due to the low-volume, high complexity nature of these projects, it is challenging to predict manufacturing times accurately [1]. Currently, the construction industry predicts manufacturing times based on the experience of shop managers. This approach, however, is prone to errors. These errors lead to inaccurate predictions, causing manufacturing schedules to become ineffective [10].

These ineffective manufacturing schedules result in an unbalanced workload between subsequent manufacturing steps. Due to the unbalanced workload, the flow of products through the manufacturing process becomes disrupted, leading to buffers between manufacturing steps.

The past decade, the construction industry started using Building Information Models (BIM). In these models information about the construction project is stored. The BIM is shared between all collaborators in the project, stimulating cooperation and reducing errors due to miscommunication [7].

Concurrently, data analysis is becoming an indespensable technique for manufacturing processes [4].

In line with these developments, Hu et al. [10] proposed a Multiple Linear Regression (MLR) prediction model based on historical data. For this prediction model, physical properties extracted from BIM are coupled to the manufacturing time. This approach turned out to improve the prediction accuracy significantly, compared to the current experience based prediction approach. The MLR prediction model, however, only yields accurate results if the relationship between physical properties and manufacturing times shows linear behavior.

In this research a general prediction model is proposed for the prediction of manufacturing times using (historical) information from (BIM) and the manufacturing process. The proposed prediction model is validated in a case study conducted at Oostingh Staalbouw, a company responsible for the design, manufacturing and assembly phase of complex steel structures.

In the remainder of this paper, the available data for this case study is discussed in section II. The relevant literature is reviewed in section III. The proposed prediction model is elaborated in section IV and experiments in order to verify and validate the proposed prediction model are discussed in section V. In section VI this paper is completed with a discussion and recommendations for further research.

II. Available data

The available data for this research are the physical properties of the products extracted from BIM (input variables) and the manufacturing time per manufacturing step (output variable). Both quantitative and categorical physical properties can be identified.

The quantitative physical properties are:

- Total weight of the product
- Maximum length of the product
- Total weld length
- Number of parts
- Number of welds
- Number of holes

Spearman's correlation coefficient is used to determine whether a relationship exists between these quantitative physical properties and manufacturing times. The found correlation coefficients vary between 0.65 and 0.92, implying that a relationship exists. The drawback of this approach, however, is that it only shows that a monotonic relationship exists, but it remains ambiguous whether this relationship is linear or nonlinear [19].

The categorical physical properties distinguished are the different profile types for the products:

- H-beam
- U-Beam
- Hollow section
- Plate

It is expected that handling of these profile types differs slightly. Therefore the prediction model should be able to incorporate these categorical input variables, along with quantitative input variables.

Since the manufacturing process at Oostingh Staalbouw has significant human involvement, human related uncertainty in the data is present. Therefore, the prediction model should be robust against uncertainty in the data.

Based on the available data, the prediction model should be able to:

- Predict continuous variables
- Cope with linear and nonlinear relationships
- Have quantitative and categorical input variables
- Be able to deal with uncertainty in the data
- Be easily interpretable in order to increase the rate of acceptance in the conservative construction industry

III. Literature

Literature on the prediction of manufacturing times using physical properties extracted from BIM in combination with historical data is limited to one study by Hu et al. [10]. In this study, a Multiple Linear Regression (MLR) prediction model is proposed. This approach resulted in significantly more accurate predictions of manufacturing times compared to the current, experience based approach common in the construction industry.

In other industries, the main focus of research is on the prediction of product lead time rather than product manufacturing time per manufacturing step. The main difference between these two is that product lead time is defined as the *"The time required once the product* *began its manufacture until the time it is completely processed"* [3], while the manufacturing time per manufacturing step is the time it takes for the product to get processed at one manufacturing step.

In these studies, several common prediction models are recognised: Multiple Linear Regression (MLR), Support Vector Regression (SVR), Tree Based Regression (TBR), and Neural Networks (NN). An overview of the implemented prediction models in related literature is provided in table 1.

Table 1: *Overview of implemented models in reviewed literature*

Reference	**Year**	**Industry**	**MLR**	**TBR**	**SVR**	**NN**
Tirkel [24]	2013	Semiconductor				X
Hu et al. [10]	2014	Prefabrication	X			
Pfeiffer et al. [21]	2015		X	X		
Hur et al. [11]	2015	Shipbuilding	X	X		
Yu and Cai [26]	2015	Aircraft				X
Mohsenijam and Lu [18]	2016	Prefabrication	X			
Arash et al. [2]	2017	Construction	X			
Lingitz et al. [17]	2018	Semiconductor	X	X	X	X
Nagahara and Nonaka [20]	2018	Semiconductor	X			
Gyulai et al. [9]	2018	Optical	X	X	X	

Along with the prediction models used in the related literature, Quinlan [22] proposed the Linear Model Tree (LMT). A LMT prediction model constructs a tree, similar to TBR, but rather than discrete values, it contains MLR models in the leaves. This way, nonlinear relationships can be broken down into multiple linear relationships.

Using the prerequisites determined in section II, the different prediction models are compared. Based on this comparison, an opportunity for an improved prediction model is found by combining the LMT and SVR prediction models.

Instead of MLR prediction models in the nodes of the prediction tree, SVR prediction models can be used. Since SVR is known to outperform MLR if uncertainty in the data is present, it is expected that the proposed adaptation is more robust against uncertainty in the data [6]. This proposed adaptation to the LMT will be named a Support Vector Regression Model Tree (SVRMT) prediction model.

IV. Support Vector Regression Model Tree

Basically, the construction of the SVRMT is analogue to the LMT as discussed by Quinlan [22] and Wang and Witten [25].

The construction of the SVRMT consists of several steps; at first the data is split in linear segments. Afterwards, a SVR prediction model is placed at each leaf. To reduce the chance of overfitting, the tree is pruned. In order to compensate for sharp discontinuities between adjacent leaves, the predictions are smoothed.

Splitting

Since the construction of the optimal prediction tree is NP-complete [12], heuristics are commonly used. For the LMT, a greedy search is performed in order to get to a sub-optimal tree. During this greedy search, for each node the input variables are sort in ascending order. For each possible split value of the predictor variable, the data is split [14]. For both the left and right side of the split, a linear model is built. The standard deviation of the residuals for both sides of the split are derived. The standard deviation reduction compared to the situation without a split (linear model in the parental node) is calculated using equation 1.

$$SDR = sd(T) - \sum_{i} \left(\frac{|T_i|}{|T|} * sd(T_i) \right) \quad (1)$$

With T the parental node and T_i the respective child nodes. This process is repeated until the number of data in the nodes is smaller than the number of input variables, or until the SDR is smaller than 0.05 times the standard deviation of the parental node [25].

Modelling in the nodes

As stated in the introduction of this section, the main difference between LMT and SVRMT prediction models is in the prediction model used in the nodes of the model tree. Instead

Table 2: *Comparison of prediction models using prerequisites based on data characteristics*

	Output function	Input variables	Accurate for	Robust against uncertainty	Interpretability
MLR	Continuous	Quantitative	Linear	No	Good
SVR	Continuous	Quantitative	Linear	Yes	Good
TBR	Discrete	Quantitative / Categorical	Linear / Nonlinear	No	Good
NN	Continuous	Quantitative / Categorical	Linear / Nonlinear	No	Bad
LMT	Continuous	Quantitative / Categorical	Linear / Nonlinear	No	Good
SVRMT	Continuous	Quantitative / Categorical	Linear / Nonlinear	Yes	Good

of MLR models, the SVRMT model uses SVR models in its nodes.

Instead of aiming to find a line with least deviation from the data points, like MLR, SVR attempts to find the narrowest tube around a function f(x), that has at most ϵ deviation from this function f(x). Data points outside this tube are penalized with factor C. This could be written as the optimization problem, described in equation 2 [6].

$$\arg\min\left\{\frac{1}{2}||w||^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*)\right\} \tag{2}$$

Subject to

$$\begin{aligned} y_i - \{w, x_i\} - b &\leq \epsilon + \xi_i \\ \{w, x_i\} + b - y_i &\leq \epsilon + \xi_i \\ \xi_i, \xi_i^* &\geq 0 \end{aligned}$$

The subset of input variables used in the SVR model is found using greedy selection; one by one, the input variables are left out. As soon as dropping variables results in increased expected error, the subset of input variables is found.

For the determination of the best combination of C and ϵ a grid search method[8] is used in combination with 5-fold cross validation [15].

Pruning

Since Tree Based prediction models are known to be prone to overfitting, precautions need to be taken [13]. In order to reduce the chance of overfitting the model, a pruning strategy can be used. During the pruning of a SVRMT, the expected error of parental nodes is compared to the expected error of both left and right child of the node. In order to get the expected error, the error of the mean absolute error of the data in the node is multiplied by a factor $\frac{n+v}{n-v}$, with n the number of training data reaching the node and v the number of input variables. The expected error can be derived using equation 3.

$$Error = \frac{n+v}{n-v} * \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}| \tag{3}$$

If the expected error of the parental node is smaller than the error in the child nodes, the parental node is pruned and becomes a leaf. This process is repeated, from bottom to top, for all nodes in the tree.

Predicting using SVRMT

In order to compensate for possible sharp discontinuities between adjacent leaves, a smoothing procedure can be used. At first the corresponding leaf for the value to be predicted is found by descending the decision tree. The corresponding model in the leaf is used to predict the value.

Afterwards, the tree is ascended from leaf to root. At each node passed, a new prediction q is made using the prediction model in the node. The smoothed prediction p′ is made using equation 4

$$p' = \frac{np + kq}{n + k} \tag{4}$$

Where p is the prediction passed from the child node, n the number of data points reaching the child node. k is a constant smoothing parameter (Wang and Witten [25] suggests k=15).

V. Experiments

The verification and validation of the proposed SVRMT prediction model is performed in a series of experiments. In addition, MLR, SVR, TBR and LMT prediction models are tested in these experiments in order to compare the proposed prediction model with prediction models used in literature.

The Mean Absolute Percentage Error (MAPE) is a common performance indicator to evaluate the prediction accuracy of prediction models. Since this indicator uses the relative error, it can be used to compare results between different studies [5]. The equation to compute the MAPE is shown in equation 5.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)/y_i| \tag{5}$$

Where y is the real manufacturing time, $\hat{y}$ is the predicted manufacturing time and n is the number of products predicted.

Unless stated otherwise, for each experiment the data is split into a training and test set using a 70/30 ratio. Each experiment is repeated tenfold, using different random seeds to split the data set.

i. Verification

In order to verify the assumption that the proposed SVRMT prediction model functions satisfactory under the prerequisites set for the prediction model, several theoretical experiments are conducted. The proposed SVRMT prediction model is tested for the following relationships:

- Linear relationship
- Quadratic relationship
- Step relationship

Heteroskedastic noise is added to these relationships in order to verify the assumption that the proposed model is capable of reconstructing relationships with uncertainty in the data.

The evaluated relationships are shown in Figure 1.

Results

The resulting MAPE of the experiments for the different prediction models is shown in table 3. For the linear relationship, the proposed SVRMT prediction model yields equal results to the SVR prediction model. This implies that the SVRMT prediction model consists solely of a root with a SVR prediction model. For nonlinear relationships (Quadratic and Step), the SVRMT prediction model outperforms the other evaluated prediction models in terms of prediction accuracy.

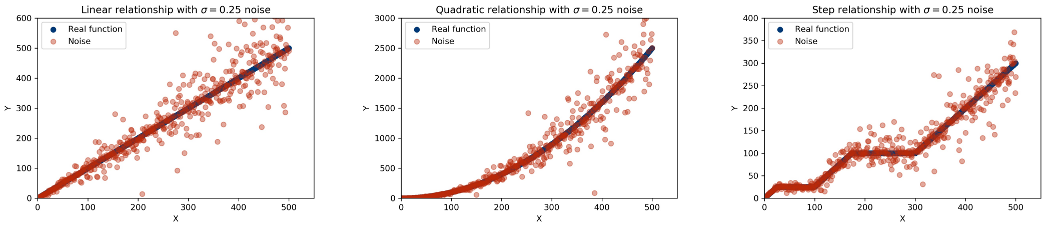
Table 3: *MAPE for the evaluated prediction models in the verification phase*

	Linear	**Quadratic**	**Step**
MLR	0.47	9.24	0.22
SVR	0	7.15	0.24
TBR	0.10	0.36	0.17
LMT	0.04	0.10	0.13
SVRMT	0	0.06	0.11

ii. Validation

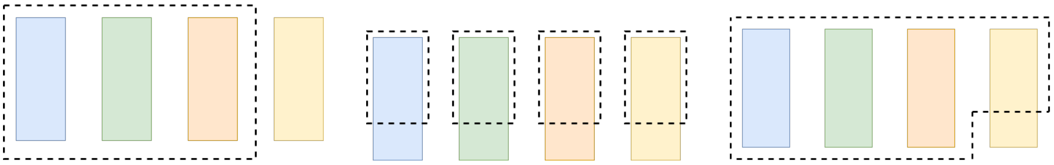
In order to validate the proposed prediction model, the model is tested in a case study. For this case study, three scenarios corresponding to different phases in the manufacturing of a construction project are studied (Figure 2). Real manufacturing data is provided by Oostingh Staalbouw. This data consists of products manufactured for four projects in the period September 2018-June 2019.

This research is limited to the assembly and welding step in the manufacturing process of Oostingh Staalbouw. Currently, it is common for the construction industry to predict the manufacturing time of a group of products (for example per truck load), rather than to predict the manufacturing time per product [10]. In 2017, Oostingh Staalbouw conducted an experiment to predict the manufacturing time per product. From this experiment, a MAPE of 0.60 can be derived. It is assumed that this value is still applicable for this study.



(a) **(b)** **(c)**

Figure 1: *Respectively the a) linear, b) quadratic and c) step relationships with added noise*



(a) **(b)** **(c)**

Figure 2: *Respectively a) Scenario 1, b) Scenario 2 and c) Scenario 3. Each colored box represents data of a construction project. The dashed lined box represents the available training data for the prediction model.*

In order to gain insight in the possible prediction accuracy, a benchmark is set. For this benchmark, all repetitive products are predicted using the mean of the specific, repetitive product. The MAPE for this strategy for the evaluated projects is 0.25. This strategy, however, can not be applied for the prediction of new, unique products. Therefore, it is expected that the MAPE will be higher for the prediction of manufacturing times of new products. For this study, an objective MAPE of 0.30 has been used.

Scenario 1

In the first scenario, the manufacturing time of products for a new construction project are predicted. The available information is limited to realized projects by the company. This scenario corresponds to the start of the manufacturing phase of a new construction project.

This scenario is evaluated for each project being left out once. Since this train / test split is not random based, this experiment is performed once.

Scenario 2

The second scenario is after a substantial part of the products for a construction project are manufactured. For convenience, 70% is used.

Scenario 3

The last scenario used to validate the proposed SVRMT prediction model corresponds to the same stage as scenario 2. In scenario 3, however, data of realized projects is available, along with data of realized products from the project to be predicted.

Results

The results of the experiments for the various scenarios are summarized in table 4 and table 5.

Notable is the difference between the linear models (MLR and SVR) and the nonlinear models (LMT and SVRMT) for the prediction of assembly times. The nonlinear models yield significantly more accurate results than the linear models, implying that the relationship between physical properties and assembly time is nonlinear. In addition, the proposed SVRMT

Table 4: *Averaged MAPE over the evaluated projects by the compared prediction models for the assembly step*

	Scenario 1	Scenario 2	Scenario 3
MLR	0.74 ± 0.37	0.48 ± 0.12	0.53 ± 0.09
SVR	0.63 ± 0.32	0.44 ± 0.17	0.43 ± 0.05
TBR	0.66 ± 0.18	0.49 ± 0.09	0.45 ± 0.04
LMT	0.47 ± 0.05	0.41 ± 0.09	0.42 ± 0.01
SVRMT	0.41 ± 0.04	0.38 ± 0.10	0.38 ± 0.01

prediction model yields both most accurate and constant results. In scenario 1, less accurate predictions are yielded for projects significantly different, in terms of physical properties, from the projects used as training data.

The accuracy of the predictions for the assembly step increases from scenario 1 to scenario 2. Predictions in scenario 2 turned out to be increasingly accurate with increasing ratio of repetitive products in the construction project. On the other hand, predictions were less accurate for relatively small construction projects with a small ratio of repetitive products. In scenario 3, both positive and negative outliers across the projects were flattened out, leading to most constant results.

Table 5: *Averaged MAPE over the evaluated projects by the compared prediction models for the welding step*

	Scenario 1	Scenario 2	Scenario 3
MLR	0.43 ± 0.05	0.42 ± 0.09	0.41 ± 0.03
SVR	0.40 ± 0.04	0.44 ± 0.11	0.38 ± 0.04
TBR	0.54 ± 0.10	0.48 ± 0.12	0.44 ± 0.11
LMT	0.44 ± 0.09	0.40 ± 0.09	0.38 ± 0.03
SVRMT	0.40 ± 0.02	0.37 ± 0.09	0.38 ± 0.03

For the welding step, more constant results across different prediction models are found. The small difference in terms of accuracy between the various prediction models (with the exception of the TBR model) implies that the relationship between the physical properties and the welding time is linear.

Overall, all implemented prediction models have increased accuracy over the current, experience based predictions. The proposed SVRMT prediction model yields slightly more accurate predictions over the prediction models used in comparable studies.

VI. Conclusion

Currently, it is common for the construction industry to predict manufacturing times of products based on experience of shop managers. This approach is prone to errors, leading to ineffective manufacturing schedules. In this paper, a new approach for predicting manufacturing times per manufacturing step based on (historical) data from Building Information Models (BIM) and the manufacturing process has been proposed.

In this paper, a combination of Linear Model Trees (LMT) and Support Vector Regression (SVR) models has been proposed. It was expected that the Support Vector Regression Model Tree (SVRMT) was able to yield accurate predictions for both linear and nonlinear relationships under influence of uncertainty in the data.

During verification of the SVRMT prediction model, it was shown that the prediction model meets this expectation. Afterwards, the SVRMT prediction model was validated in three real case scenarios. For these real case scenarios, data from Oostingh Staalbouw has been used.

Compared to the current, experience based approach the proposed SVRMT prediction model turned out to be a significant improvement. The Mean Absolute Percentage Error was reduced from 0.60 up to 0.37.

Even though, the objective MAPE of 0.30 was not met, the proposed prediction model is a signification improvement compared to the current approach. Based on this research, recommendations for further research can be identified.

During this research, process related input variables were not taken into account for the prediction model. It is expected that embedding more relevant input variables will increase the accuracy of the prediction model. In addition it would be interesting to enlarge the training data with more projects carried out by the company, which will likely decrease the variance of residuals [23].

In this research, a standard top-down approach is used, while Kretowski and Czajkowski [16] proposed an evolutionary approach for the construction of model trees. It would therefore be interesting to compare Model Trees constructed using various heuristics to find the optimal prediction model.

Furthermore, the application of the prediction model can be extended to the design phase of construction projects. Based on the manufacturing time prediction model, more sophisticated design choices can be made to optimize products in terms of manufacturing costs.

Lastly, the proposed SVRMT prediction model should be tested in different manufacturing processes (in the construction industry) to elaborate on the generality of the prediction model.

References

[1] Federico Adrodegari, Andrea Bacchetti, Roberto Pinto, Fabiana Pirola, and Massimo Zanardini. Engineer-to-order (ETO) production planning and control: an empirical framework for machinery-building companies. *Prod. Plan. Control*, 26 (11):910–932, aug 2015. ISSN 0953-7287. doi: 10.1080/09537287.2014.1001808. URL https://doi.org/10.1080/09537287.2014.1001808.

[2] Mohsenijam Arash, Siu Ming-Fung Francis, and Lu Ming. Modified Stepwise Regression Approach to Streamlining Predictive Analytics for Construction Engineering Applications. *J. Comput. Civ. Eng.*, 31(3):04016066, may 2017. doi: 10.1061/(ASCE)CP.1943-5487.0000636. URL https://doi.org/10.1061/(ASCE)CP.1943-5487.0000636.

[3] F T S Chan. Performance Measurement in a Supply Chain. *Int. J. Adv. Manuf. Technol.*, 21(7):534–548, may 2003. ISSN 1433-3015. doi: 10.1007/s001700300063. URL https://doi.org/10.1007/s001700300063.

[4] Ying Cheng, Ken Chen, Hemeng Sun, Yongping Zhang, and Fei Tao. Data and knowledge mining with big data towards smart production. *J. Ind. Inf. Integr.*, 9:1–13, mar 2018. ISSN 2452-414X. doi: 10.1016/J.JII.2017.08.001. URL https://www.sciencedirect.com/science/article/pii/S2452414X17300584{#}fig0002.

[5] Arnaud de Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean Absolute Percentage Error for regression models. *Neurocomputing*, 192:38–48, 2016. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2015.12.114. URL http://www.sciencedirect.com/science/article/pii/S0925231216003325.

[6] Harris Drucker, Christopher J C Burges, Linda Kaufman, Alex J Smola, and Vladimir Vapnik. Support vector regression machines. In *Adv. Neural Inf. Process. Syst.*, pages 155–161, 1997.

[7] Chuck Eastman, Paul Teicholz, Rafael Sacks, and Kathleen Liston. *BIM handbook: A guide to building information modeling for owners, managers, designers, engineers and contractors*. John Wiley & Sons, 2011. ISBN 111802169X.

[8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[9] Dávid Gyulai, András Pfeiffer, Gábor Nick, Viola Gallina, and Wilfried Sihn. Lead time prediction in a flow-shop environment with analytical and machine learning approaches. *IFAC-PapersOnLine*, 51(11):1029–1034, jan 2018. ISSN 2405-8963. doi: 10.1016/J.IFACOL.2018.08.472. URL https://www.sciencedirect.com/science/article/pii/S2405896318316008?via{%}3Dihub.

[10] X Hu, M Lu, and S Abourizk. BIM-based data mining approach to estimating job man-hour requirements in structural steel fabrication. In *Proc. - Winter Simul. Conf.*, volume 2015-January, pages 3399–3410, 2015. doi: 10.1109/WSC.2014.7020173. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84940519162{&}doi=10.1109{%}2FWSC.2014.7020173{&}partnerID=40{&}md5=af32c2a66c7e6ba57b594e6a50c93b59.

[11] Minhoe Hur, Seung-kyung Lee, Bongseok Kim, Sungzoon Cho, Dongha Lee, and Daehyung Lee. A study on the man-hour prediction system for shipbuilding. *J. Intell. Manuf.*, 26(6):1267–1279, 2015. ISSN 1572-8145. doi: 10.1007/s10845-013-0858-3. URL https://doi.org/10.1007/s10845-013-0858-3.

[12] Laurent Hyafil and Ronald L Rivest. Constructing optimal binary decision trees is NP-complete. *Inf. Process. Lett.*, 5(1):15–17, 1976.

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An introduction to statistical learning*, volume 112. Springer, New York, NY, New York, 2013. ISBN 978-1-4614-7138-7. doi: https://doi.org/10.1007/978-1-4614-7138-7.

[14] Aram Karalič. Employing Linear Regression in Regression Tree Leaves. In *Proc. 10th Eur. Conf. Artif. Intell.*, ECAI '92, pages 440–441, New York, NY, USA, 1992. John Wiley & Sons, Inc. ISBN 0-471-93608-1. URL http://dl.acm.org/citation.cfm?id=145448.146775.

[15] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[16] Marek Kretowski and Marcin Czajkowski. An Evolutionary Algorithm for Global Induction of Regression Trees. In Rutkowski Leszek, , Rafał Scherer, and Tadeusiewicz Ryszard, and Zadeh Lotfi A., and and Zurada Jacek M, editors, *Artifical Intell. Soft Comput.*, pages 157–164, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-13232-2.

[17] Lukas Lingitz, Viola Gallina, Fazel Ansari, Dávid Gyulai, András Pfeiffer, Wilfried Sihn, and László Monostori. Lead time prediction using machine learning algorithms: A case study by a semiconductor manufacturer. *Procedia CIRP*, 72:1051–1056, jan 2018. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2018.03.148. URL https://www.sciencedirect.com/science/article/pii/S2212827118303056?via{%}3Dihub.

[18] Arash Mohsenijam and Ming Lu. Achieving Sustainable Structural Steel Design by Estimating Fabrication Labor Cost Based on BIM Data. *Procedia Eng.*, 145:654–661, jan 2016. ISSN 1877-7058. doi: 10.1016/J.PROENG.2016.04.056. URL https://www.sciencedirect.com/science/article/pii/S1877705816300613?via{%}3Dihub.

[19] M M Mukaka. Statistics corner: A guide to appropriate use of correlation coefficient in medical research. *Malawi Med. J.*, 24(3):69–71, sep 2012. ISSN 1995-7270. URL https://www.ncbi.nlm.nih.gov/pubmed/23638278https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3576830/.

[20] Satoshi Nagahara and Youichi Nonaka. Product-specific Process Time Estimation from Incomplete Point of Production Data for Mass Customization. *Procedia CIRP*, 67:558–562, jan 2018. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2017.12.260. URL https://www.sciencedirect.com/science/article/pii/S2212827117312064?via{%}3Dihub.

[21] András Pfeiffer, Dávid Gyulai, Botond Kádár, and László Monostori. Manufacturing Lead Time Estimation with the Combination of Simulation and Statistical Learning Methods. *Procedia CIRP*, 41:75–80, jan 2016. ISSN 2212-8271. doi: 10.1016/J.PROCIR.2015.12.018. URL https://www.sciencedirect.com/science/article/pii/S2212827115010975?via{%}3Dihub.

[22] John R Quinlan. Learning with continuous classes. In *5th Aust. Jt. Conf. Artif. Intell.*, volume 92, pages 343–348. World Scientific, 1992.

[23] Justin D. Salciccioli, Yves Crutain, Matthieu Komorowski, and Dominic C. Marshall. *Sensitivity Analysis and Model Validation*, pages 263–271. Springer International Publishing, Cham, 2016. ISBN 978-3-319-43742-2. doi: 10.1007/978-3-319-43742-2_17. URL https://doi.org/10.1007/978-3-319-43742-2_17.

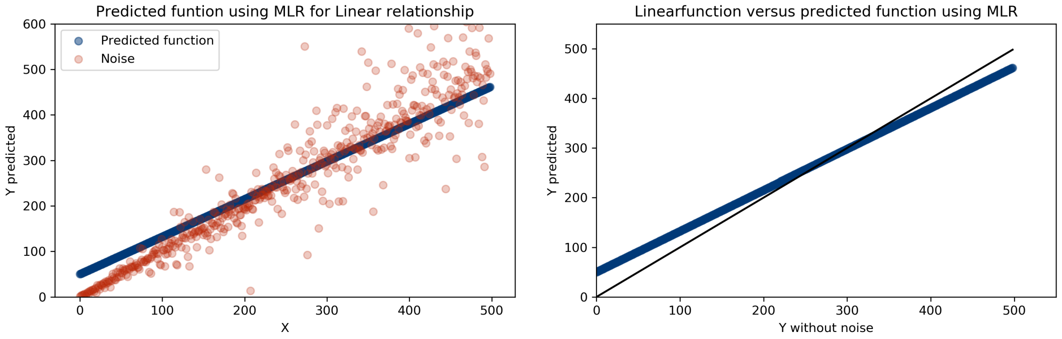
[24] Israel Tirkel. Forecasting flow time in semiconductor manufacturing using knowledge discovery in databases. *Int. J. Prod. Res.*, 51(18):5536–5548, sep 2013. ISSN 0020-7543. doi: 10.1080/00207543.2013.787168. URL https://doi.org/10.1080/00207543.2013.787168.

[25] Yong Wang and Ian H Witten. Induction of model trees for predicting continuous classes. Technical report, 1996. URL https://hdl.handle.net/10289/1183.

[26] T Yu and H Cai. The prediction of the man-hour in aircraft assembly based on support vector machine particle swarm optimization. *J. Aerosp. Technol. Manag.*, 7(1):19–30, 2015. doi: 10.5028/jatm.v7i1.409. URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-84924762303{&}doi=10.5028{%}2Fjatm.v7i1.409{&}partnerID=40{&}md5=518371d17a5e838ac5f637da354699e4.

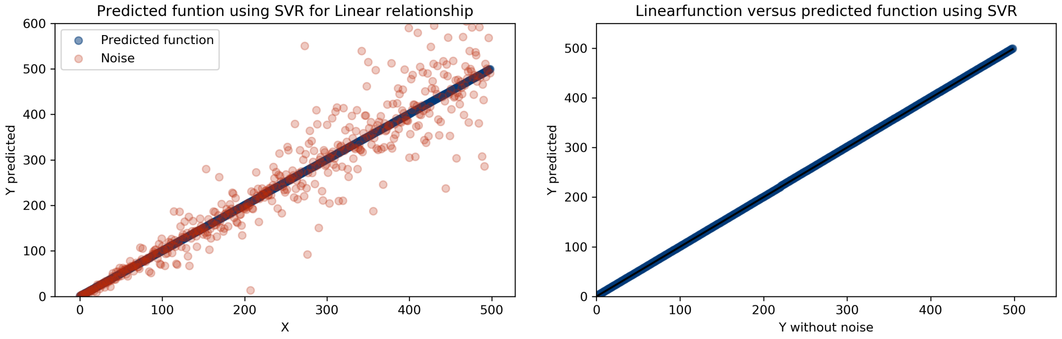
B
Verification

B.1. Linear


Predicted funtion using MLR for Linear relationship

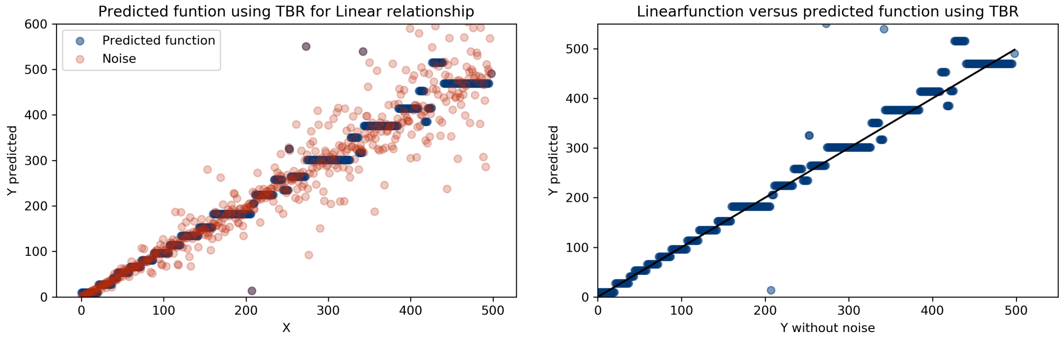
Linearfunction versus predicted function using MLR

a) Reconstructed linear relationship by the MLR prediction model b)inverse fitted value plot linear relationship MLR prediction model

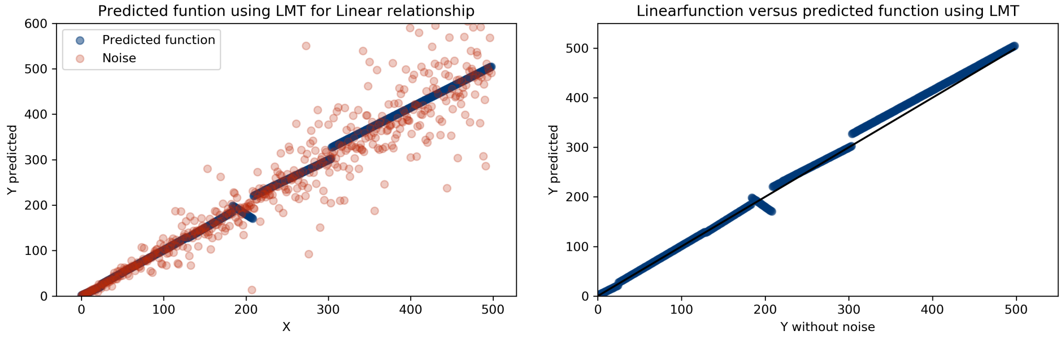

Predicted funtion using SVR for Linear relationship

Linearfunction versus predicted function using SVR

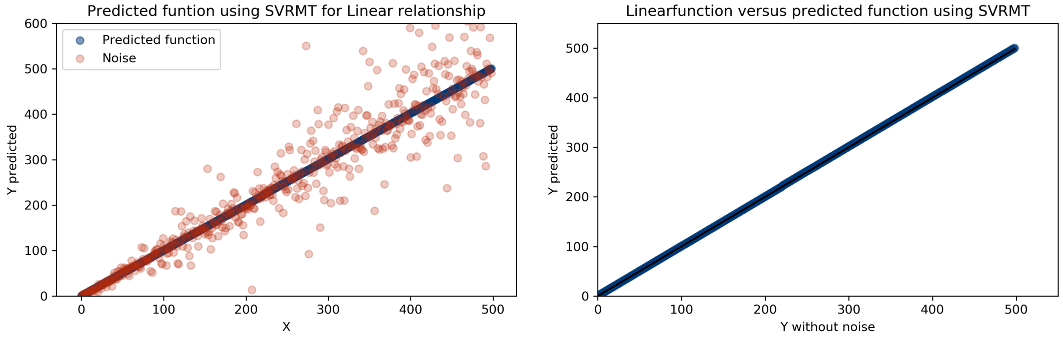
a) Reconstructed linear relationship by the SVR prediction model b)inverse fitted value plot linear relationship SVR prediction model



a) Reconstructed linear relationship by the TBR prediction model b)inverse fitted value plot linear relationship TBR prediction model

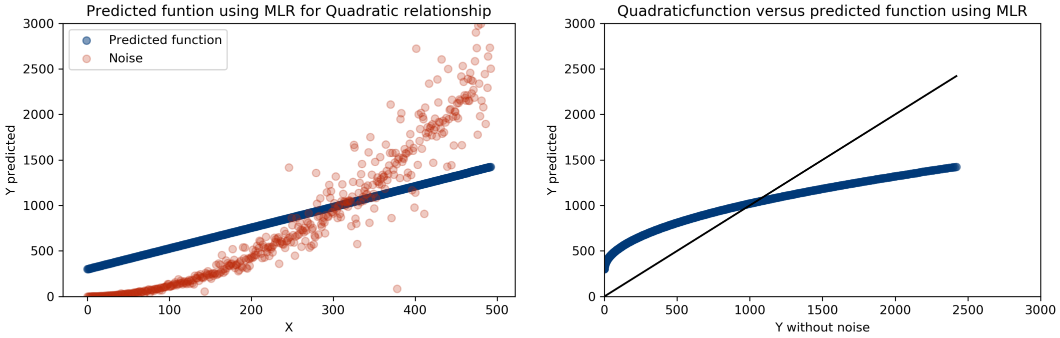


a) Reconstructed linear relationship by the LMT prediction model b)inverse fitted value plot linear relationship LMT prediction model

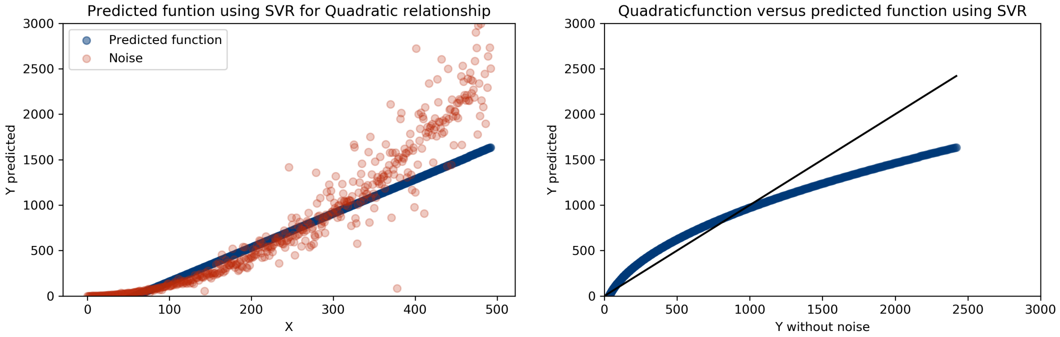


a) Reconstructed linear relationship by the SVRMT prediction model b)inverse fitted value plot linear relationship SVRMT prediction model

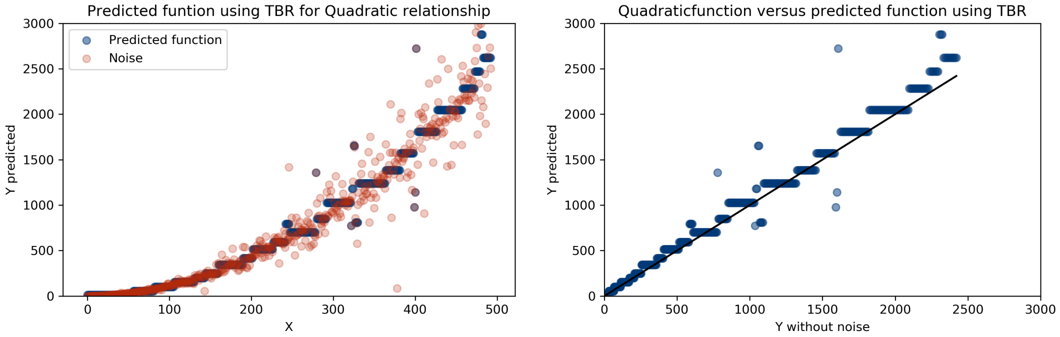
B.2. Quadratic



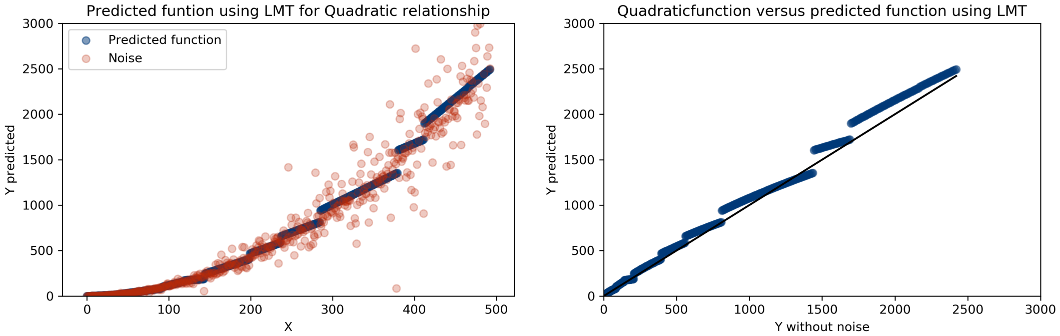
a) Reconstructed Quadratic relationship by the MLR prediction model b)inverse fitted value plot Quadratic relationship MLR prediction model



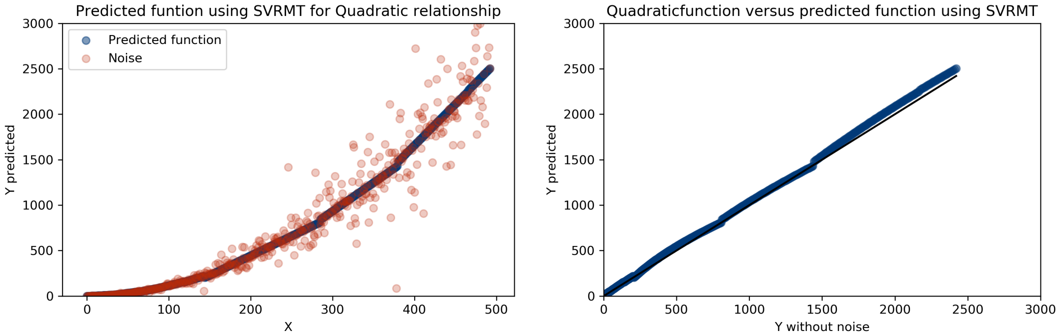
a) Reconstructed Quadratic relationship by the SVR prediction model b)inverse fitted value plot Quadratic relationship SVR prediction model



a) Reconstructed Quadratic relationship by the TBR prediction model b)inverse fitted value plot Quadratic relationship TBR prediction model

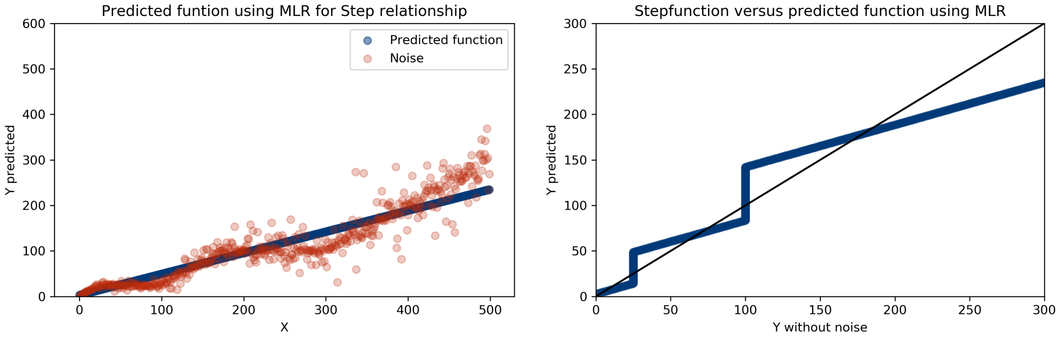


a) Reconstructed Quadratic relationship by the LMT prediction model b)inverse fitted value plot Quadratic relationship LMT prediction model

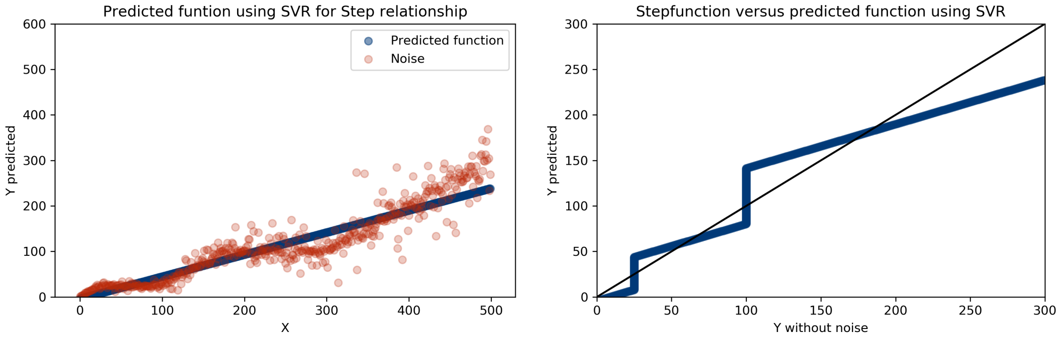


a) Reconstructed Quadratic relationship by the SVRMT prediction model b)inverse fitted value plot Quadratic relationship SVRMT prediction model

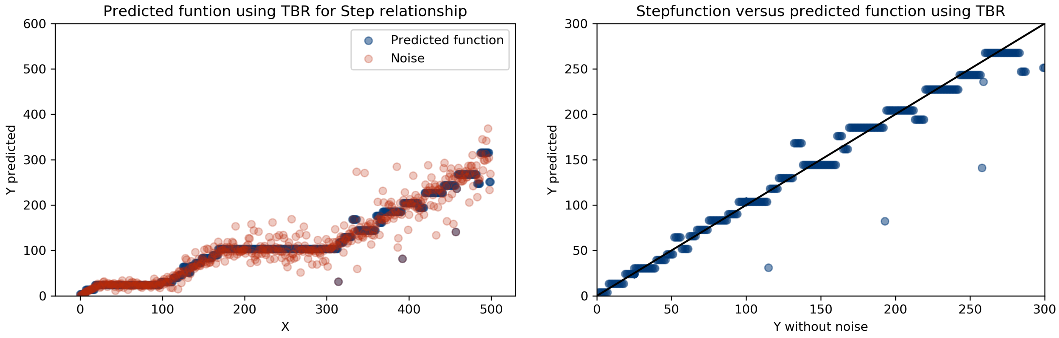
B.3. Step



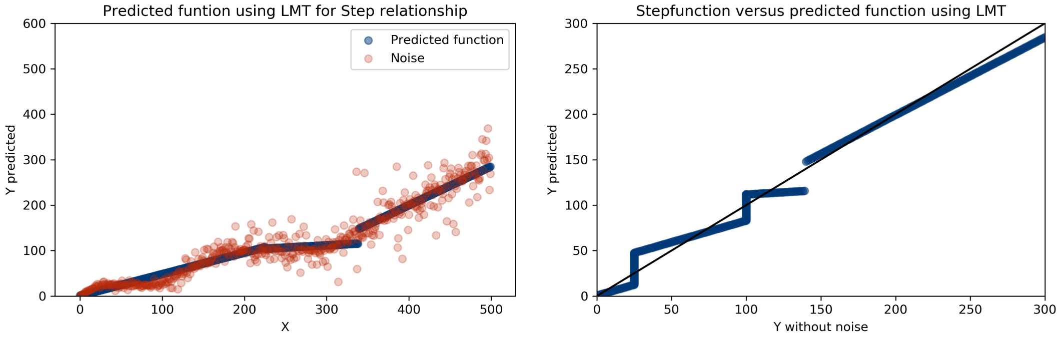
a) Reconstructed Step relationship by the MLR prediction model b)inverse fitted value plot Step relationship MLR prediction model



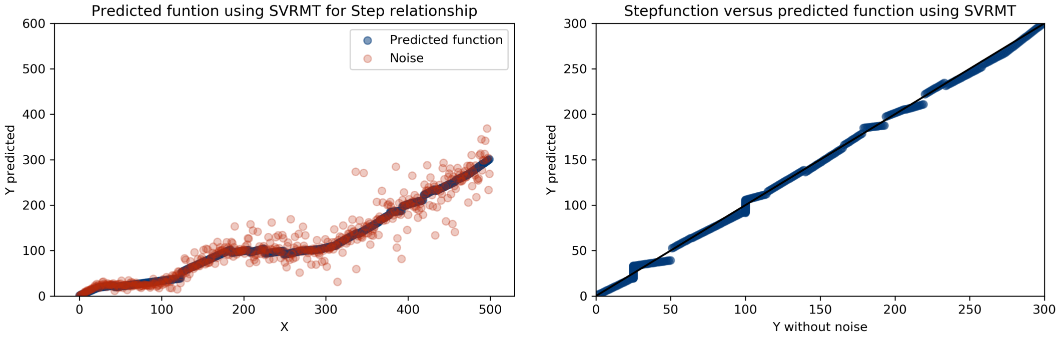
a) Reconstructed Step relationship by the SVR prediction model b)inverse fitted value plot Step relationship SVR prediction model



a) Reconstructed Step relationship by the TBR prediction model b)inverse fitted value plot Step relationship TBR prediction model



a) Reconstructed Step relationship by the LMT prediction model b)inverse fitted value plot Step relationship LMT prediction model



a) Reconstructed Step relationship by the SVRMT prediction model b)inverse fitted value plot Step relationship SVRMT prediction model

C

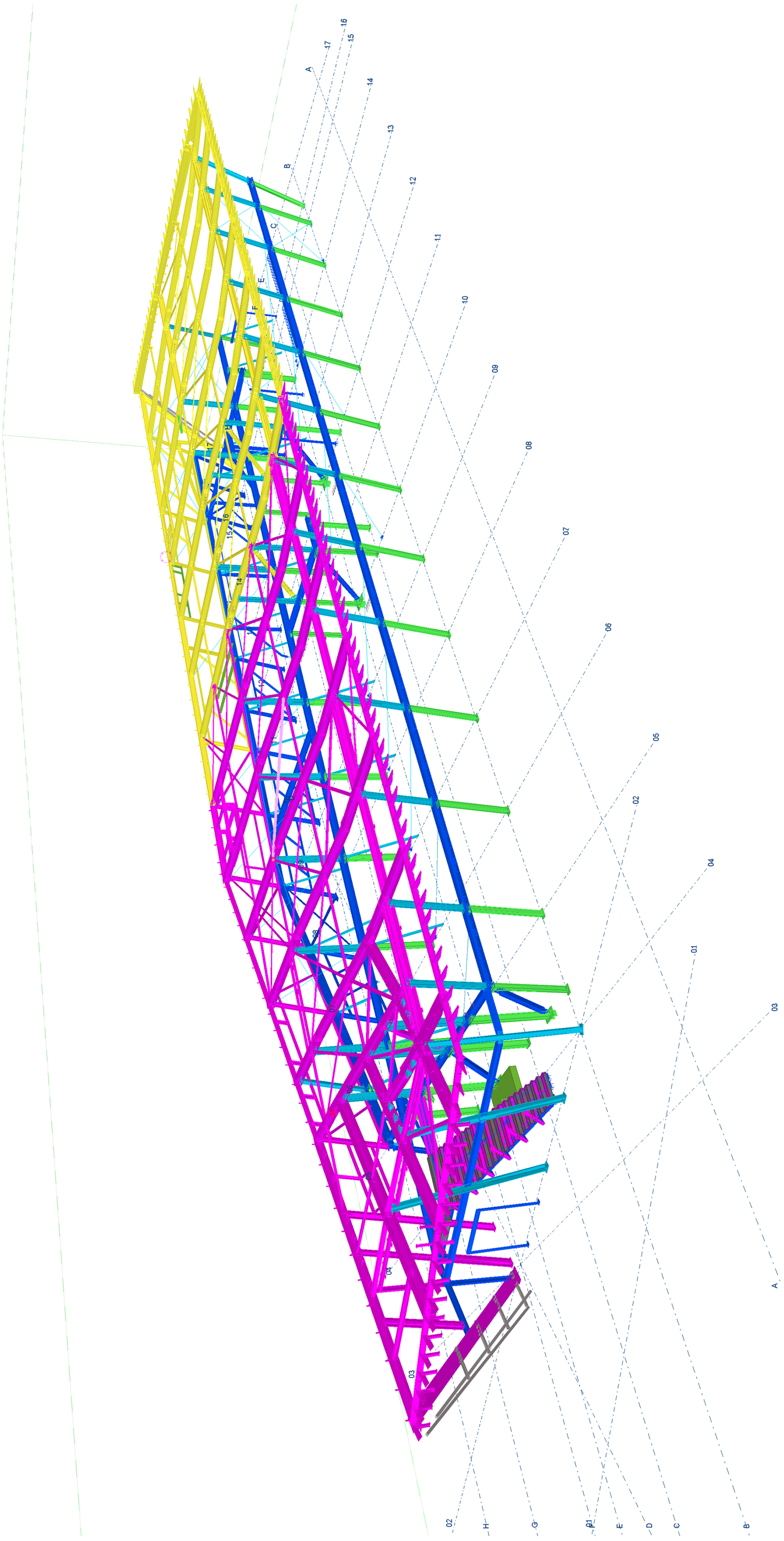
Validation

C.1. ProjectSummary

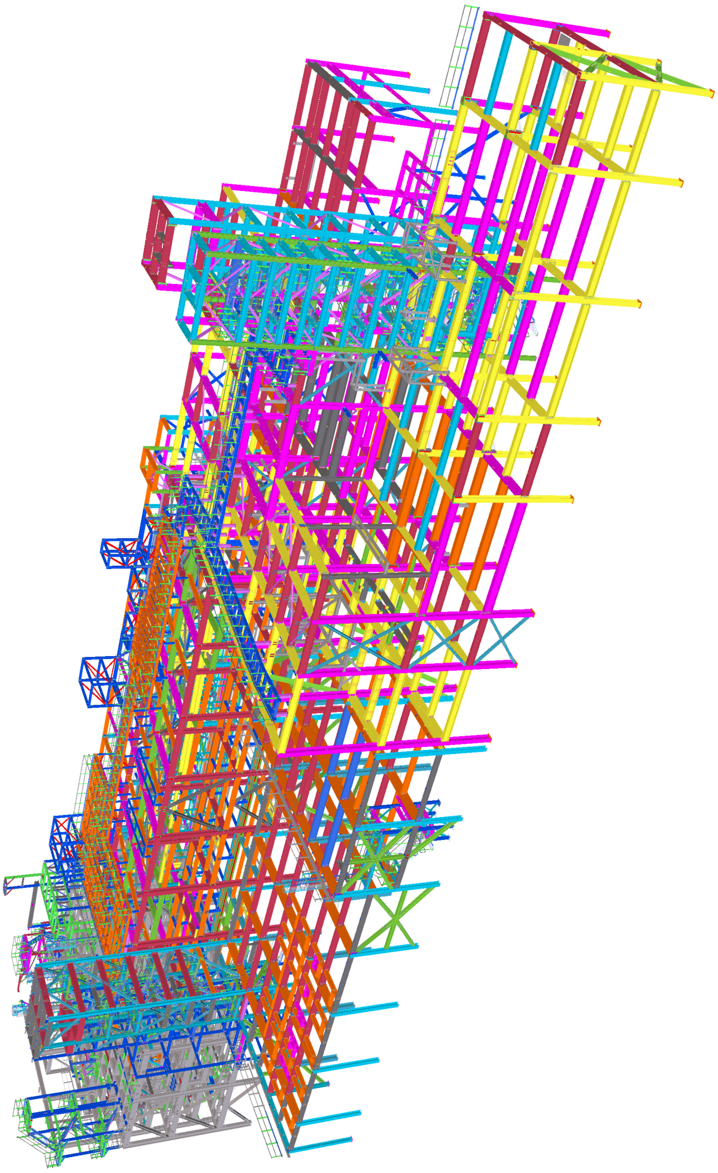
Summary of properties evaluated in this research

	2048	181017	184062	194003
Size	504	4568	996	407
Ratio Unique Products	0,64	0,65	0,72	0,10
Mean Assembly Time [h]	3,30	2,00	2,52	1,75
Median Assembly Time [h]	1,47	1,17	1,44	0,97
Std Assembly Time [h]	4,63	2,20	2,82	2,54
Max Assembly Time [h]	30,28	28,65	20,17	15,62
Mean Weld Time [h]	2,46	3,35	2,57	2,24
Median Weld Time [h]	1,34	1,77	1,53	1,85
Std Weld Time [h]	2,72	4,35	4,04	1,35
Max Weld Time [h]	15,05	43,26	34,88	8,70
No. Parts mean	7,10	6,39	7,27	6,39
No. Parts median	6,00	5,00	6,00	7,00
No. Parts std	3,83	4,07	4,44	2,09
No. Welds mean	5,20	17,41	11,69	10,05
No. Welds median	2,00	14,00	8,00	6,00
No. Welds std	10,18	13,60	13,64	6,46
Weight [kg] mean	708,94	770,68	747,83	373,42
Weight [kg] median	427,50	443,00	591,00	180,00
Weight [kg] std	746,81	847,26	958,44	450,37
Length [mm] mean	6576,10	5310,42	8835,13	3962,77
Length [mm] median	4886,00	4887,00	7500,00	2790,00
Length [mm] std	4539,13	3463,99	4555,25	2292,71
WeldLength [mm] mean	5919,01	12414,36	8409,33	5479,57
WeldLength [mm] median	3336,00	5273,50	6009,50	3380,00
WeldLength [mm] std	9459,52	16826,03	7601,24	4683,68
H	0,50	0,92	0,24	0,29
K	0,47	0,03	0,73	0,60
U	0,01	0,05	0,03	0,00
P	0,02	0,00	0,00	0,04

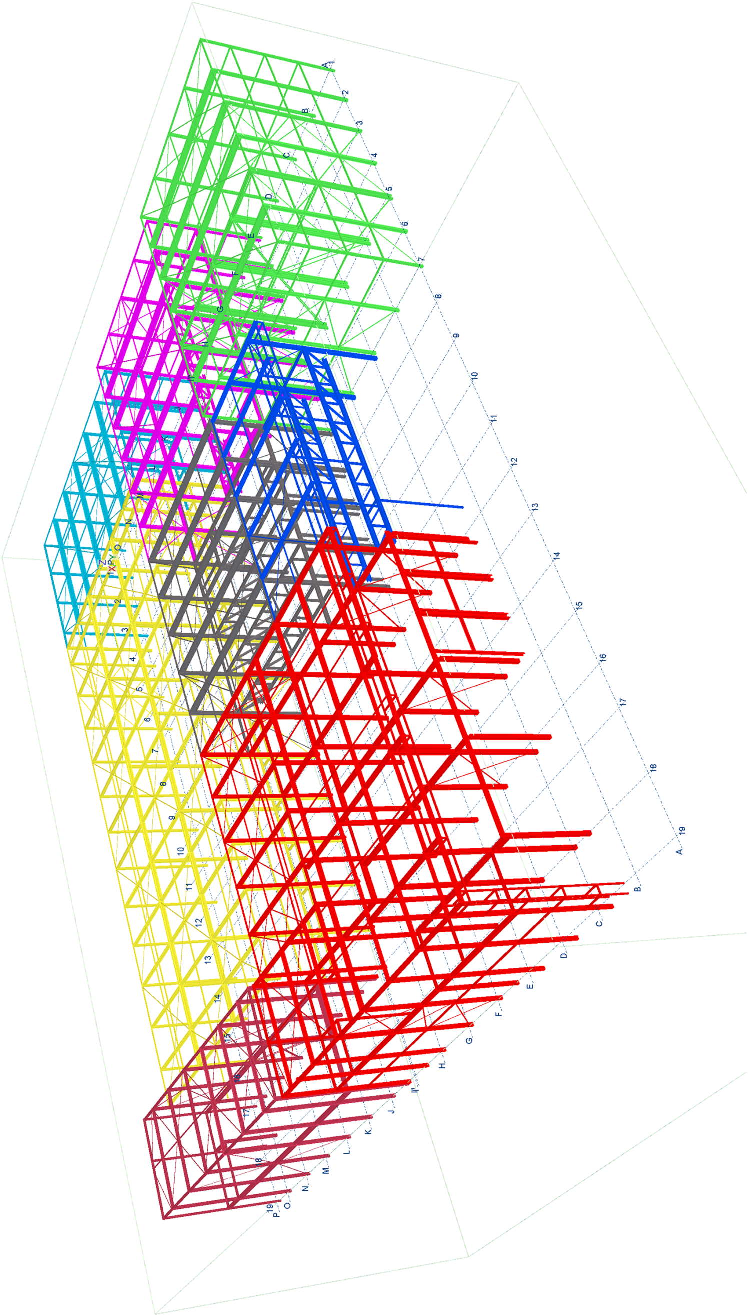
C.1.1. Project 2048



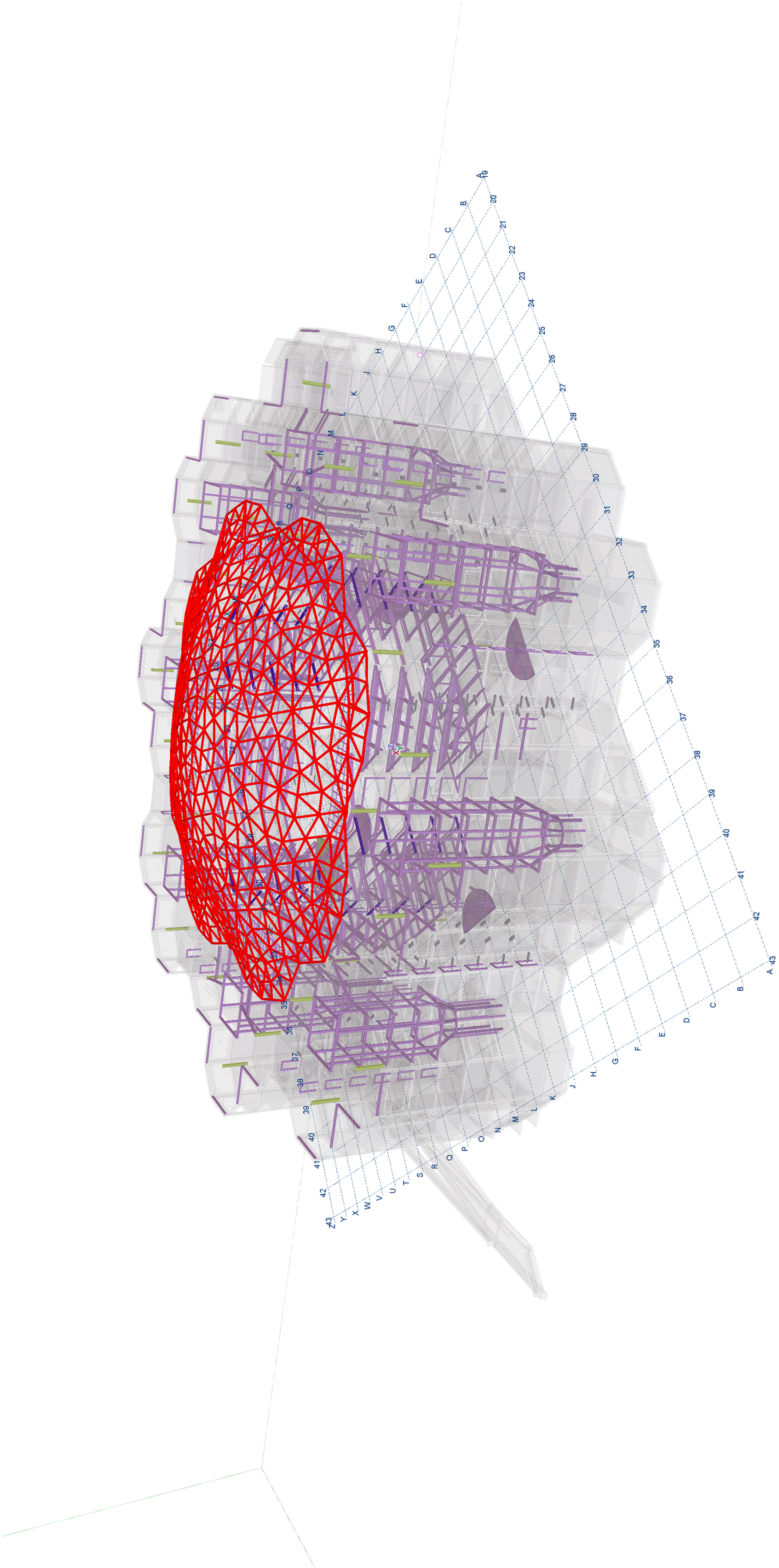
C.1.2. Project 181017



C.1.3. Project 184062



C.1.4. Project 194003



C.2. Tables

C.2.1. Scenario 1

2048

Results of Scenario1 for 2048 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.45	0.41	0.33	0.92
SVR	0.45	0.43	0.31	0.92
TBR	0.72	0.6	0.69	0.74
LMT	0.43	0.39	0.31	0.92
SVRMT	0.43	0.41	0.3	0.93

Results of Scenario1 for 2048 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.45	0.4	0.31	0.85
SVR	0.44	0.4	0.28	0.86
TBR	0.53	0.41	0.61	0.67
LMT	0.41	0.36	0.27	0.88
SVRMT	0.42	0.37	0.26	0.88

184062

Results of Scenario1 for 184062 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.52	0.38	0.55	0.74
SVR	0.34	0.28	0.31	0.91
TBR	0.42	0.32	0.5	0.81
LMT	0.46	0.33	0.58	0.75
SVRMT	0.34	0.27	0.32	0.9

Results of Scenario1 for 184062 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.51	0.43	0.41	0.78
SVR	0.42	0.33	0.33	0.85
TBR	0.63	0.39	0.71	0.53
LMT	0.56	0.45	0.43	0.74
SVRMT	0.42	0.35	0.33	0.85

181017

Results of Scenario1 for 181017 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.61	0.44	0.61	0.47
SVR	0.56	0.49	0.48	0.61
TBR	0.91	0.47	1.43	-1.06
LMT	0.44	0.33	0.55	0.64
SVRMT	0.41	0.32	0.47	0.72

Results of Scenario1 for 181017 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.36	0.3	0.32	0.94
SVR	0.34	0.29	0.29	0.95
TBR	0.62	0.42	0.73	0.76
LMT	0.36	0.3	0.32	0.94
SVRMT	0.36	0.3	0.32	0.94

194003

Results of Scenario1 for 194003 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	1.36	1.17	0.92	-1.95
SVR	1.16	0.9	0.94	-1.52
TBR	0.57	0.31	0.72	0.04
LMT	0.56	0.42	0.7	0.12
SVRMT	0.45	0.36	0.49	0.51

Results of Scenario1 for 194003 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.44	0.31	0.4	0.41
SVR	0.36	0.27	0.34	0.59
TBR	0.39	0.27	0.42	0.46
LMT	0.34	0.27	0.3	0.65
SVRMT	0.41	0.34	0.31	0.46

C.2.2. Scenario 2

181017

Results of Scenario2 for 181017 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.46	0.35	0.42	0.72
SVR	0.45	0.36	0.39	0.74
TBR	0.43	0.31	0.52	0.67
LMT	0.39	0.31	0.35	0.8
SVRMT	0.39	0.31	0.34	0.81

Results of Scenario2 for 181017 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.4	0.28	0.42	0.91
SVR	0.4	0.28	0.42	0.91
TBR	0.41	0.3	0.47	0.9
LMT	0.39	0.27	0.4	0.92
SVRMT	0.37	0.28	0.36	0.93

2048

Results of Scenario2 for 2048 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.68	0.4	0.8	0.71
SVR	0.7	0.43	0.92	0.64
TBR	0.61	0.35	0.81	0.67
LMT	0.56	0.4	0.6	0.83
SVRMT	0.52	0.38	0.57	0.84

Results of Scenario2 for 2048 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.56	0.36	0.64	0.64
SVR	0.58	0.4	0.68	0.55
TBR	0.63	0.4	0.85	0.32
LMT	0.54	0.36	0.59	0.69
SVRMT	0.51	0.38	0.52	0.73

184062

Results of Scenario2 for 184062 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.39	0.28	0.39	0.86
SVR	0.38	0.28	0.35	0.87
TBR	0.54	0.37	0.66	0.64
LMT	0.39	0.28	0.39	0.86
SVRMT	0.36	0.27	0.36	0.88

Results of Scenario2 for 184062 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.39	0.32	0.35	0.86
SVR	0.35	0.29	0.28	0.9
TBR	0.54	0.36	0.67	0.23
LMT	0.4	0.32	0.34	0.86
SVRMT	0.35	0.27	0.3	0.89

194003

Results of Scenario2 for 194003 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.37	0.31	0.28	0.76
SVR	0.24	0.2	0.19	0.9
TBR	0.37	0.24	0.43	0.55
LMT	0.31	0.26	0.23	0.83
SVRMT	0.24	0.2	0.19	0.9

Results of Scenario2 for 194003 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.31	0.21	0.29	0.68
SVR	0.29	0.2	0.27	0.71
TBR	0.32	0.22	0.32	0.68
LMT	0.28	0.2	0.27	0.73
SVRMT	0.26	0.2	0.24	0.77

C.2.3. Scenario 3

2048

Results of Scenario3 for 2048 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.47	0.39	0.41	0.89
SVR	0.43	0.4	0.3	0.92
TBR	0.47	0.35	0.51	0.86
LMT	0.41	0.37	0.3	0.93
SVRMT	0.4	0.37	0.26	0.94

Results of Scenario3 for 2048 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.45	0.38	0.43	0.8
SVR	0.44	0.38	0.34	0.85
TBR	0.57	0.39	0.75	0.47
LMT	0.36	0.31	0.3	0.89
SVRMT	0.4	0.35	0.29	0.88

184062

Results of Scenario3 for 184062 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.45	0.32	0.46	0.81
SVR	0.36	0.28	0.38	0.87
TBR	0.48	0.31	0.63	0.68
LMT	0.44	0.35	0.43	0.83
SVRMT	0.38	0.28	0.38	0.87

Results of Scenario3 for 184062 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.42	0.34	0.31	0.86
SVR	0.38	0.33	0.28	0.89
TBR	0.49	0.34	0.57	0.69
LMT	0.43	0.35	0.31	0.86
SVRMT	0.41	0.35	0.31	0.86

181017

Results of Scenario3 for 181017 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.51	0.38	0.48	0.65
SVR	0.42	0.33	0.39	0.77
TBR	0.45	0.33	0.57	0.62
LMT	0.42	0.32	0.37	0.78
SVRMT	0.36	0.29	0.33	0.83

Results of Scenario3 for 181017 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.38	0.28	0.4	0.92
SVR	0.37	0.27	0.37	0.93
TBR	0.41	0.3	0.47	0.9
LMT	0.36	0.27	0.35	0.94
SVRMT	0.36	0.27	0.34	0.94

194003

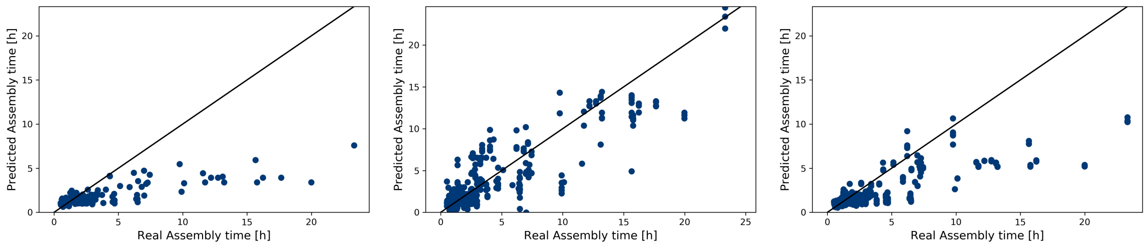
Results of Scenario3 for 194003 Assembly

Assembly	MAPE	MeAPE	Std	R^2
MLR	0.69	0.63	0.49	0.22
SVR	0.49	0.4	0.44	0.52
TBR	0.38	0.27	0.48	0.56
LMT	0.41	0.34	0.35	0.67
SVRMT	0.38	0.33	0.3	0.74

Results of Scenario3 for 194003 Weld

Weld	MAPE	MeAPE	Std	R^2
MLR	0.37	0.25	0.35	0.51
SVR	0.34	0.26	0.3	0.67
TBR	0.27	0.19	0.25	0.78
LMT	0.36	0.25	0.34	0.61
SVRMT	0.34	0.27	0.3	0.67

C.3. Inverse Fitted Value Plots Validation

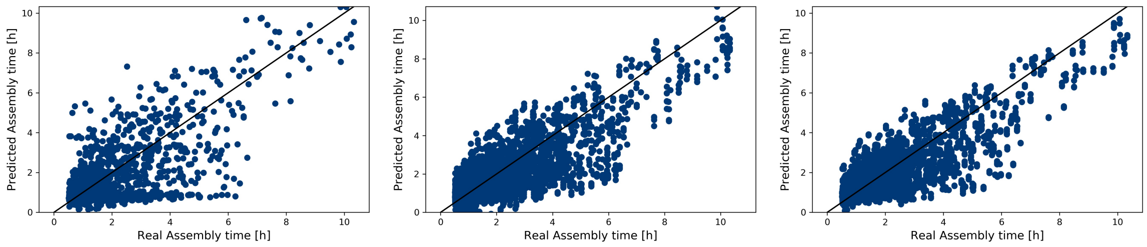


(a)

(b)

(c)

Inverse fitted value plot for the predicted welding times for project 181017 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model

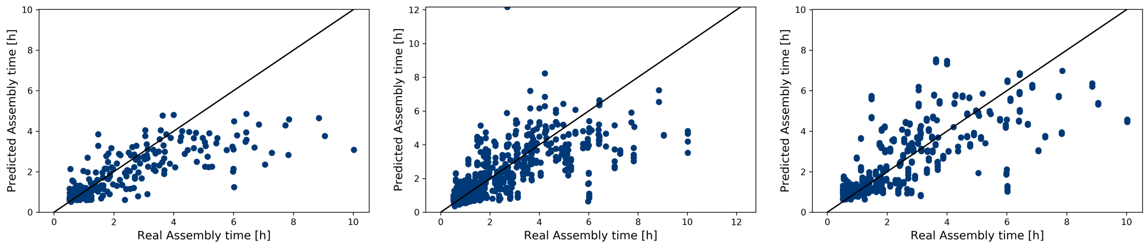


(a)

(b)

(c)

Inverse fitted value plot for the predicted assembly times for project 181017 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model

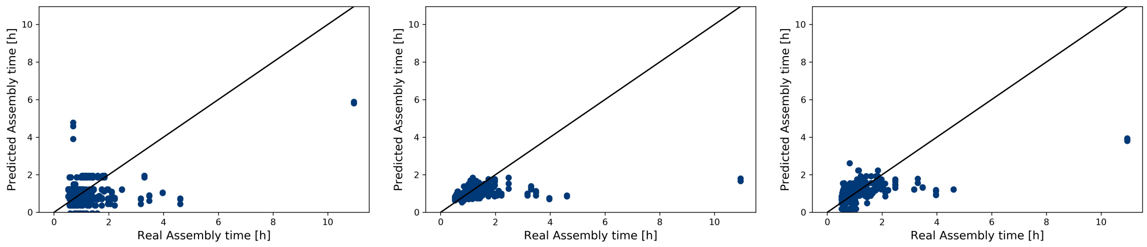


(a)

(b)

(c)

Inverse fitted value plot for the predicted assembly times for project 184062 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model

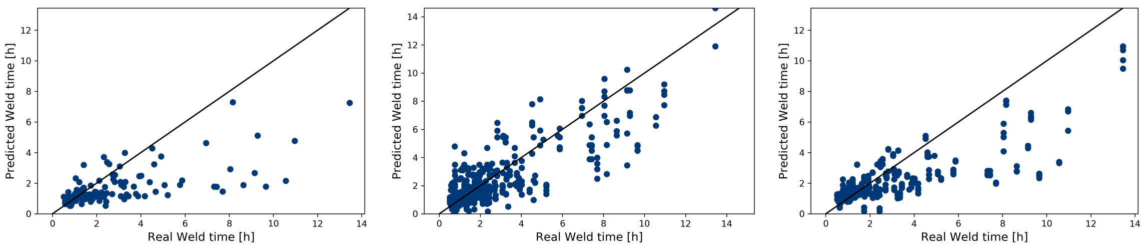


(a)

(b)

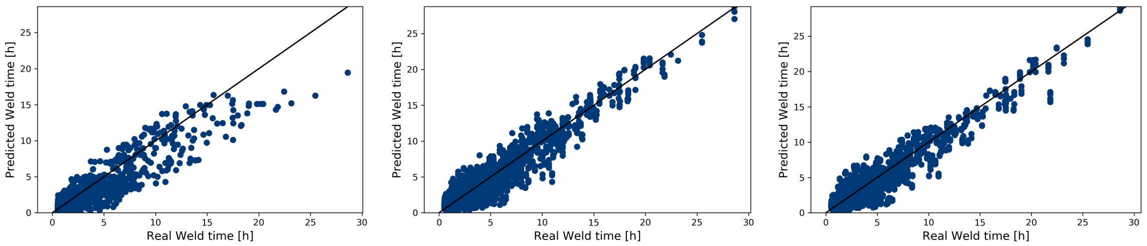
(c)

Inverse fitted value plot for the predicted assembly times for project 194003 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model



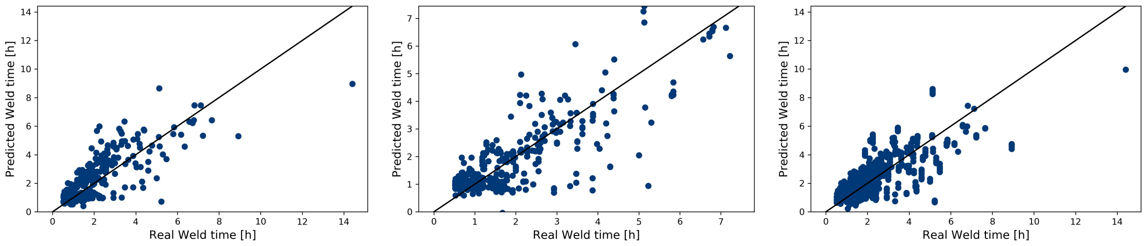
(a) (b) (c)

Inverse fitted value plot for the predicted Weld times for project 2048 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model



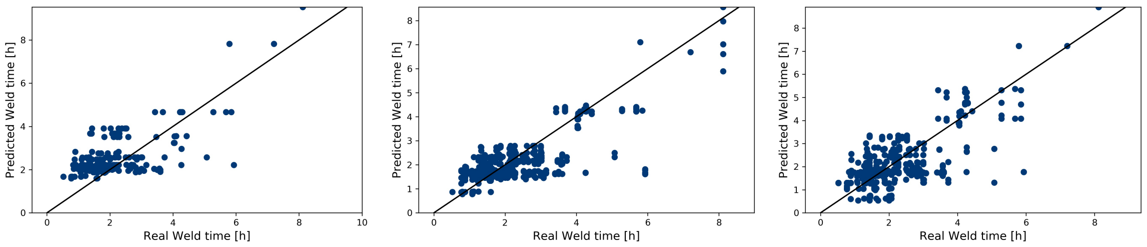
(a) (b) (c)

Inverse fitted value plot for the predicted Weld times for project 181017 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model



(a) (b) (c)

Inverse fitted value plot for the predicted Weld times for project 184062 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model



(a) (b) (c)

Inverse fitted value plot for the predicted Weld times for project 194003 in a) scenario 1, b) scenario 2 and c) scenario 3 using the **SVRMT** prediction model

D

Python Code

```
# -*- coding: utf-8 -*-
"""
Created on Wed Jun  5 10:53:13 2019

@author: l.vanderplas
"""

import pandas as pd
import numpy as np
import time
import copy
import math
from sklearn import linear_model
from sklearn import svm
from sklearn.feature_selection import RFECV
from sklearn.svm import SVR
from sklearn.tree import DecisionTreeRegressor

from sklearn.model_selection import GridSearchCV, train_test_split
from sklearn.preprocessing import RobustScaler, StandardScaler
from pandas import ExcelWriter
from pandas import ExcelFile

global AllColumns
global Continuous
global YColumn
global AllNodes

#Predict for all products in Test Set
def ResultatenZoeker(i, TestSet, TreeType):
    SmallResults = pd.DataFrame()
    for index, row in TestSet.iterrows():
        X = TestSet.loc[[index], AllColumns]
        YTest = TestSet.loc[[index], YColumn].values
        YPred, leaf = TreeType.search(X)
#          SmallResults.at[i, 'X'] = X
        SmallResults.at[str(i)+'-'+str(index), 'Y'] = YTest
        SmallResults.at[str(i)+'-'+str(index), 'YPred'] = YPred
        SmallResults.at[str(i)+'-'+str(index), 'Res'] = YTest - YPred
        SmallResults.at[str(i)+'-'+str(index), 'RelRes'] = (YTest - YPred)/YTest
        SmallResults.at[str(i)+'-'+str(index), 'AbsRes'] = np.abs((YTest - YPred))
        SmallResults.at[str(i)+'-'+str(index), 'AbsRelRes'] = np.abs((YTest - YPred)/YTest)

    return SmallResults

#Compute Performance Indicators for Test Set
def Metrics(SmallResults, Runtime, i):
    SmallMetrics = pd.DataFrame()
    SmallMetrics.at[i,'MAPE'] = SmallResults['AbsRelRes'].mean()
    SmallMetrics.at[i,'MeAPE'] = SmallResults['AbsRelRes'].median()
    SmallMetrics.at[i,'Std'] = SmallResults['AbsRelRes'].std()
    SmallMetrics.at[i,'R2'] = 1-(np.square(SmallResults['YPred'] - SmallResults['Y']).sum()/(np.square(SmallResults['Y']-
         SmallResults['Y'].mean()).sum()))
    SmallMetrics.at[i,'Runtime␣[s]'] = Runtime

    return SmallMetrics

#Combine Results with Test Set
def Resultaten(XTest, YTest, YPred):
    SmallResults = XTest
    SmallResults['Y'] = YTest
    SmallResults['YPred'] = YPred
    SmallResults['Res'] = YTest - YPred
    SmallResults['RelRes'] = (YTest - YPred)/YTest
    SmallResults['AbsRes'] = np.abs((YTest - YPred))
    SmallResults['AbsRelRes'] = np.abs((YTest - YPred)/YTest)

    return SmallResults

#Build MLR Prediction Model
def RAWMLR(TrainSet, TestSet):
    XTrain = TrainSet[Continuous]
    YTrain = TrainSet[YColumn]
    XTest = TestSet[Continuous]
    YTest = TestSet[YColumn]

    scaler = RobustScaler().fit(XTrain)
    XTrainScaled = scaler.transform(XTrain)
    XTestScaled = scaler.transform(XTest)
    reg = linear_model.ElasticNetCV(cv=5).fit(XTrainScaled,YTrain)
    YPred = reg.predict(XTestScaled)

    SmallResults = Resultaten(XTest, YTest, YPred)

    return SmallResults

#Build SVR Prediction Model
def RAWSVR(TrainSet, TestSet):
    parameters = [{'C': [0.1, 1, 10,25], 'epsilon': [0.1, 0.5, 1]}]
    ColumnSet = []

    Verbetering = True

    XTrain = TrainSet[Continuous]
    YTrain = TrainSet[YColumn]

    XTest = TrainSet[Continuous]
    YTest = TrainSet[YColumn]

    Scaler = RobustScaler().fit(XTrain)
    XTrainScaled = Scaler.transform(XTrain)
    XTestScaled = Scaler.transform(XTest)

    Reg = GridSearchCV(svm.LinearSVR(), parameters, cv=5)
    Reg.fit(XTrainScaled,YTrain)

    if len(XTrain.columns) == 1:
        YPred = Reg.predict(XTestScaled)
        SmallResults = Resultaten(XTest, YTest, YPred)
        return SmallResults

    #Greedy Backwards Elimation
    else:
        YPred = Reg.predict(XTrainScaled)
        Residuals = np.abs(YTrain-YPred)

        n = len(XTrain)
        v = len(Continuous)
        factor = (n+v)/(n-v)

        Error = factor*Residuals.mean()

        BestColumn = Continuous
        BestReg = Reg
        BestScaler = Scaler

        YTrain = TrainSet[YColumn]
        YTest = TestSet[YColumn]

        while Verbetering:
            Verbetering = False
            for column in Continuous:
                if column not in ColumnSet:
                    SmallColumn = ColumnSet.copy()
                    SmallColumn.append(column)

                    X = TrainSet[SmallColumn]
                    Scaler = RobustScaler().fit(X)
                    Xscaled = Scaler.transform(X)

                    Reg = GridSearchCV(svm.LinearSVR(max_iter=-1), parameters, cv=5)
                    Reg.fit(Xscaled,YTrain)
```

```python
                    Y_pred = Reg.predict(Xscaled)
                    Residuals = np.abs(YTrain-Y_pred)

                    n = len(X)
                    v = len(SmallColumn)
                    factor = (n+v)/(n-v)

                    SmallError = factor*Residuals.mean()
                    if SmallError <= Error:
                        Verbetering = True
                        Error = SmallError
                        BestColumn = column
                        BestReg = Reg
                        BestScaler = Scaler

            if Verbetering:
                ColumnSet.append(BestColumn)

        if len(ColumnSet) == 0:
            ColumnSet = Continuous

        XTest = TestSet[BestColumn]

        if type(XTest) == pd.core.series.Series:
            XTest = XTest.values.reshape(-1,1)

        XTestScaled = BestScaler.transform(XTest)
        YPred = BestReg.predict(XTestScaled)
        SmallResults = Resultaten(XTest, YTest, YPred)

        return SmallResults

#Define Node Object for Model Tree
class Node():
    #Initialize
    def __init__(self, DataSet, Parent, Name, Depth):
        self.Parent = Parent
        self.DataSet = DataSet
        self.Name = Name
        self.Leaf = False
        self.Depth = Depth
        self.prune = True
        self.Left_child = None
        self.Right_child = None


    def GreedySplit(self, NodeList):
        Verbetering = False
        SDR = 0

        Xset = self.DataSet
        Yset = self.DataSet[YColumn]

        YPredSet = RAWMLR(Xset[Continuous], Yset)
        Residuals = (Yset - YPredSet)/Yset
        STDAll = np.std((Residuals))

        #Go through all input variables
        for column in AllColumns:
            TussenTijd = time.time()
            Set = Xset[column].drop_duplicates().sort_values()

            #For each unique value in input variable
            for i in (Set):
                YLeft = Xset.loc[Xset[column] <= i][YColumn]         #Define Left Set
                XLeft  = Xset.loc[Xset[column] <= i][Continuous]

                YRight = Xset.loc[Xset[column] > i][YColumn]         #Define Right Set
                XRight = Xset.loc[Xset[column] > i][Continuous]

                if len(XLeft) > MinSize and len(XRight) > MinSize:
                    YPredLeft = RAWMLR(XLeft, YLeft)
                    ResidualsLeft = np.std((YLeft - YPredLeft)/YLeft)

                    YPredRight = RAWMLR(XRight, YRight)
                    ResidualsRight = np.std((YRight - YPredRight)/YRight)

                    #Calculate Standard Deviation Reduction
                    SDR_small = STDAll - ((len(YLeft) * ResidualsLeft) + (len(YRight) * ResidualsRight))/len(Yset)

                    if SDR_small > SDR and SDR_small > 0.05*STDAll:
                        Verbetering = True
                        SDR = SDR_small
                        SplitValue = i
                        SplitColumn = column
            print(time.time() - TussenTijd)

        #If improved split is identified
        if Verbetering:
            print('Verbetering')
            print('Inital STD ' + str(STDAll))
            DataLeft = Xset.loc[Xset[SplitColumn] <= SplitValue]
            DataRight = Xset.loc[Xset[SplitColumn] > SplitValue]

            self.Left_child = Node(DataLeft, self, (len(AllNodes)), (self.Depth+1))      #Create Left Child
            NodeList.append(self.Left_child)
            AllNodes.append(self.Left_child)

            self.Right_child = Node(DataRight, self, (len(AllNodes)), (self.Depth+1))    #Create Right Child
            NodeList.append(self.Right_child)
            AllNodes.append(self.Right_child)

            #Appoint SplitColumn and SplitValue to the node
            self.PivotColumn = SplitColumn
            print(self.PivotColumn)
            self.PivotValue = SplitValue

            print('Verbeterde variabele: ' + str(SplitColumn))
            print('Split waarde: ' + str(SplitValue))
            print('Reduction: ' + str(SDR))

        #If the node is a leaf
        else:
            self.Leaf = True
            self.prune = False
            print('This is a leaf!')

        Tree.Constructor(NodeList)

    #Build model in node
    def Model(self, ModelType):
        #Set parameters for grid search
        parameters = [{'C': [0.01, 0.1, 0.5, 1, 10], 'epsilon': [0.01, 0.1, 0.5, 1]}]

        ColumnSet = []
        Y = self.DataSet[YColumn]

        Verbetering = True

        X = self.DataSet[Continuous]

        Scaler = RobustScaler().fit(X)
        XScaled = Scaler.transform(X)

        #Model for LMT
        if ModelType == 'MLR':
            Reg = linear_model.ElasticNetCV(cv=5)
            Reg.fit(XScaled,Y)

        #Model for SVRMT
        elif ModelType == 'SVR':
            Reg = GridSearchCV(svm.LinearSVR(max_iter=100000), parameters, cv=5)
            Reg.fit(XScaled,Y)

        Y_pred = Reg.predict(XScaled)
        Residuals = np.abs(Y-Y_pred)

        n = len(X)
        v = len(Continuous)
        if n == v:
            factor = (n+v)/(n-v+1)
        else:
            factor = (n+v)/(n-v)

        Error = factor*Residuals.mean()

        BestColumn = Continuous
        BestReg = Reg
        BestScaler = Scaler

        while Verbetering:
            Verbetering = False
            for column in Continuous:
                if column not in ColumnSet:
                    SmallColumn = ColumnSet.copy()
                    SmallColumn.append(column)

                    X = self.DataSet[SmallColumn]
                    Scaler = RobustScaler().fit(X)
                    Xscaled = Scaler.transform(X)

                    if ModelType == 'MLR':

                        Reg = linear_model.ElasticNetCV(cv=5)
                        Reg.fit(Xscaled,Y)

                    elif ModelType == 'SVR':
                        Reg = GridSearchCV(svm.LinearSVR(), parameters, cv=5)
                        Reg.fit(Xscaled,Y)

                    Y_pred = Reg.predict(Xscaled)
                    Residuals = np.abs(Y-Y_pred)

                    n = len(X)
                    v = len(SmallColumn)

                    if n == v:
                        factor = (n+v)/(n-v+1)
                    else:
                        factor = (n+v)/(n-v)

                    SmallError = factor*Residuals.mean()
                    if SmallError < Error:
                        Verbetering = True
                        Error = SmallError
                        BestColumn = column
                        BestReg = Reg
                        BestScaler = Scaler

            if Verbetering:
                ColumnSet.append(BestColumn)

        if len(ColumnSet) == 0:
            ColumnSet = Continuous

        self.model = BestReg
        self.scaler = BestScaler
        self.ColumnSet = ColumnSet

    def Prune(self):
        self.prune = False
        n = len(self.DataSet)
        v = len(self.ColumnSet)
        factor = (n+v)/(n-v)

        X = self.scaler.transform(self.DataSet[self.ColumnSet])
        Y = self.DataSet[YColumn]

        Y_pred = self.model.predict(X)

        Residuals = np.abs((Y - Y_pred)/Y)

        #Determine Error Factor Left Child
        n_left = len(self.Left_child.DataSet)
        v_left = len(self.Left_child.ColumnSet)
        factor_left = (n_left+v_left)/(n_left-v_left)

        #Predict for Left Child
        X_left = self.Left_child.scaler.transform(self.Left_child.DataSet[self.Left_child.ColumnSet])
        Y_left = self.Left_child.DataSet[YColumn]
        Y_predleft = self.Left_child.model.predict(X_left)
        Residuals_left = np.abs((Y_left - Y_predleft)/Y_left)

        #Determine Error Factor Right Child
        n_right = len(self.Right_child.DataSet)
        v_right = len(self.Right_child.ColumnSet)
        factor_right = (n_right+v_right)/(n_right-v_right)

        #Predict for Right Child
        X_right = self.Right_child.scaler.transform(self.Right_child.DataSet[self.Right_child.ColumnSet])
        Y_right = self.Right_child.DataSet[YColumn]
        Y_predright = self.Right_child.model.predict(X_right)
        Residuals_right = np.abs((Y_right - Y_predright)/Y_right)

        #Determine whether it should Prune
        if factor*Residuals.mean() < ((n_left/n) * factor_left*Residuals_left.mean() + (n_right/n) * factor_right*
              Residuals_right.mean()):
            print('Node␣' + str(self.Name) + '␣should␣be␣pruned')
            self.Leaf = True
            self.Left_child = None
            self.Right_child = None
        if self.Parent:
                return TreeType.Pruning(self.Parent)
        else:
            return

#Define Tree Class
class RMT():
    def __init__(self):
        self.root = None

    #Construct Tree
    def Constructor(self,NodeList):
        if self.root == None:    #Define Root of Tree
            self.root = Node(TrainSet,None,'Root', 0)
            AllNodes.append(self.root)
            self.root.GreedySplit(NodeList)

        #Keep splitting until all leaves are determined
        elif len(NodeList) > 0:
            cur_node = NodeList[0]
            print('-'*50)
            print('Current␣node␣' + str(cur_node.Name))
            print('Parent␣' + str(cur_node.Parent.Name))
            NodeList.remove(cur_node)
            cur_node.GreedySplit(NodeList)
        else:
            print('Building␣of␣Tree␣finished')

    #Search corresponding leaf for predicting a new value
    def search(self, X):
#        print('Searching')
        if self.root != None:
            return self._search(X, self.root)
        else:
            print('Het␣is␣de␣root!')
            return False

    def _search(self,X, cur_node):
        if cur_node.Leaf:
            if Smooth:
                Y_pred = Tree.smooth(X, cur_node, 0, 0)
            else:
                XSmall = X[cur_node.ColumnSet]
                X_transform = cur_node.scaler.transform(XSmall)

                Y_pred = cur_node.model.predict(X_transform)
            return Y_pred, cur_node.Name
        else:
            column = cur_node.PivotColumn
            value = cur_node.PivotValue
            if X[column].values <= value and cur_node.Left_child!=None:
                return self._search(X, cur_node.Left_child)
            elif X[column].values > value and cur_node.Right_child!=None:
                return self._search(X, cur_node.Right_child)

    #Smooth prediction
    def smooth(self, X, cur_node, p, n):
        if cur_node.Parent is not None:
            return self._smooth(X, cur_node, p, n)
        else:
            if p == 0:
                XSmall = X[cur_node.ColumnSet]
                X_transform = cur_node.scaler.transform(XSmall)
                q = cur_node.model.predict(X_transform)
                return q
            return p

    def _smooth(self, X, cur_node, p, n):
        if cur_node.Parent is None:
            XSmall = X[cur_node.ColumnSet]
            X_transform = cur_node.scaler.transform(XSmall)
            q = cur_node.model.predict(X_transform)
#            n = len(cur_node.DataSet)
            k = 15
            p_acc = (n*p + k*q)/(n+k)
            return p_acc

        elif cur_node.Leaf:
            XSmall = X[cur_node.ColumnSet]

            X_transform = cur_node.scaler.transform(XSmall)

            p_acc = cur_node.model.predict(X_transform)
            return self.smooth(X, cur_node.Parent, p_acc, len(cur_node.DataSet))
        else:
            XSmall = X[cur_node.ColumnSet]
            X_transform = cur_node.scaler.transform(XSmall)

            q = cur_node.model.predict(X_transform)
#             n = len(cur_node.DataSet)
            k = 15
            p_acc = (n*p + k*q)/(n+k)
            return self.smooth(X, cur_node.Parent, p_acc, len(cur_node.DataSet))

    #Build a prediction model in each node of the tree
    def InorderModel(self, cur_node, ModelType):
        print(cur_node.Name)
        ModelTijd = time.time()
        cur_node.Model(ModelType)
        if not cur_node.Leaf:
            self.InorderModel(cur_node.Left_child, ModelType)
            self.InorderModel(cur_node.Right_child, ModelType)
        ModelTijd = time.time() - ModelTijd

        return ModelTijd

    def Pruning(self, cur_node):
        if cur_node.Left_child is not None:
            if cur_node.Left_child.prune:
                self.Pruning(cur_node.Left_child)
        if cur_node.Right_child is not None:
            if cur_node.Right_child.prune:
                self.Pruning(cur_node.Right_child)
        else:
            if cur_node.Name != 'Root':
                cur_node.Prune()
        return

    #Print Final Tree
    def TreePrinter(self, cur_node, size):
        size.append(cur_node)
        if not cur_node.Leaf:
            print(cur_node.Name)
            print(cur_node.PivotColumn)
            if cur_node.PivotColumn in SplitVars:
                SplitVars[cur_node.PivotColumn] += 1
            else:
                SplitVars[cur_node.PivotColumn] = 1
            self.TreePrinter(cur_node.Left_child, size)
            self.TreePrinter(cur_node.Right_child, size)

        return size

if __name__ == '__main__':
    global MinSize
    global TrainSet
    global Smooth
    global Pruning
    global SplitVars
    global Steps
    global AllColumns

    #ResultsFrames
    ResultsMLR = pd.DataFrame()
    ResultsSVR = pd.DataFrame()
    ResultsLMT = pd.DataFrame()
    ResultsSVRMT = pd.DataFrame()

    #Detere Models to evaluate
    ModelTypes = ['RAWMLR', 'RAWSVR', 'TBR', 'LMT', 'SVRMT']

    #Determine Output Variables
    YColumns = ['Assembly␣Time␣[h]', 'Weld␣Time␣[h]']

    #Determine Input Variables
    AllColumns = ['Weight␣[kg]', 'Length␣[mm]', 'WeldLength␣[mm]', 'No.␣Parts', 'No.␣Welds', 'No.␣Holes', 'H', 'K', 'P',
        'U']
    Continuous = ['Weight␣[kg]', 'Length␣[mm]', 'WeldLength␣[mm]', 'No.␣Parts', 'No.␣Welds', 'No.␣Holes']


    AllTests = ['TotalAllFinal']

    Seeds = [1494, 2051549, 5498497, 2484, 1814, 105487, 20234234, 234,345345345, 3456789, 5117]    #Random split seeds
    Models = []
    Test = pd.DataFrame()
    TestPruning = pd.DataFrame()
    MinSizes = []
```

```
Steps = 100
Split = '194003'
Runs = 1
Scenario = 1


Smooth = True
Pruning = True

for Test in AllTests:
    print(Split)
    path = 'C:\\Users\\l.vanderplas\\Documents\\Afstuderen␣Leon\\Validation\\Results\\'

    #Set up Result Excel Sheets
    writerResultatenMLR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split +'MLR.xlsx')
    writerResultatenSVR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'SVR.xlsx')
    writerResultatenLMT = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'LMT.xlsx')
    writerResultatenSVRMT = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'SVRMT.xlsx')
    writerResultatenTBR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'TBR.xlsx')

    writerDataMLR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'RawResults' + 'MLR.xlsx')
    writerDataSVR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'RawResults' + 'SVR.xlsx')
    writerDataLMT = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'RawResults' + 'LMT.xlsx')
    writerDataSVRMT = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'RawResults' + 'SVRMT.xlsx')
    writerDataTBR = ExcelWriter(path + 'Scenario' + str(Scenario) + '-' + Split + 'RawResults' + 'TBR.xlsx')

    for YColumn in YColumns:
        SplitVars = {}
        TotalData = pd.read_excel('C:\\Users\\l.vanderplas\\Documents\\Afstuderen␣Leon\\Data\\TotalData\\'+Test+'.
              xlsx', index_col=0)
        TotalData = TotalData.loc[TotalData[YColumn] > 0.5]
        MinSizes = [len(Continuous)]

        for MinSize in MinSizes:
            ResultsMLR = pd.DataFrame()
            ResultsSVR = pd.DataFrame()
            ResultsLMT = pd.DataFrame()
            ResultsSVRMT = pd.DataFrame()
            ResultsTBR = pd.DataFrame()

            DataMLR = pd.DataFrame()
            DataSVR = pd.DataFrame()
            DataLMT = pd.DataFrame()
            DataSVRMT = pd.DataFrame()
            DataTBR = pd.DataFrame()

            for i in range(1, Runs+1):
                print(Test )
                print(YColumn)
                print(MinSize)
                print(i)
                if type(Split) == float:
                    TrainSet, TestSet = train_test_split(TotalData, test_size=Split, random_state=Seeds[i])

                else:
                    if Scenario == 1:
                        TrainSet = TotalData.loc[TotalData['Project'] != int(Split)]
                        TestSet = TotalData.loc[TotalData['Project'] == int(Split)]
                    elif Scenario == 2:
                        TotalData = TotalData[TotalData.index.str.contains(Split)]
                        TrainSet, TestSet = train_test_split(TotalData, test_size=0.3, random_state=Seeds[i])
                    elif Scenario == 3:
                        TestSet = TotalData.loc[TotalData.index.str.contains(Split)]
                        TrainSet, DummySet = train_test_split(TotalData, test_size=0.3, random_state=Seeds[i])
                        DummySet, TestSet = train_test_split(TestSet, test_size=0.3, random_state=Seeds[i])

                #Construct Model Tree
                if any(c in ModelTypes for c in ('LMT', 'SVRMT')):
                        StartTime = time.time()
                        NodeList = []
                        AllNodes = []
                        Tree = RMT()
                        ConstructionTijd = time.time()
                        Tree.Constructor(NodeList)
                        TreeTime = time.time() - StartTime

                for Model in ModelTypes:
                    #RAWMLR Code
                    if Model == 'RAWMLR':
                        StartTime = time.time()
                        SmallResults = RAWMLR(TrainSet, TestSet)
                        Runtime = time.time() - StartTime
                        SmallMetrics = Metrics(SmallResults, Runtime, i)
                    DataMLR = DataMLR.append(SmallResults)
                    ResultsMLR = ResultsMLR.append(SmallMetrics)

                #RAWSVR Code
                elif Model == 'RAWSVR':
                    StartTime = time.time()
                    SmallResults = RAWSVR(TrainSet, TestSet)
                    Runtime = time.time() - StartTime

                    SmallMetrics = Metrics(SmallResults, Runtime, i)
                    DataSVR = DataSVR.append(SmallResults)

                    ResultsSVR = ResultsSVR.append(SmallMetrics)

                #TBR Code
                elif Model == 'TBR':
                    StartTime = time.time()
                    regr_1 = DecisionTreeRegressor(max_depth=10).fit(TrainSet[AllColumns], TrainSet[YColumn])
                    Ypred = regr_1.predict(TestSet[AllColumns])
                    Runtime = time.time() - StartTime

                    SmallResults = Resultaten(TestSet[AllColumns], TestSet[YColumn], Ypred)
                    SmallMetrics = Metrics(SmallResults, Runtime, i)

                    DataTBR = DataTBR.append(SmallResults)
                    ResultsTBR = ResultsTBR.append(SmallMetrics)

                #LMT code
                elif Model == 'LMT':
                    StartTime = time.time()

                    Type = 'MLR'
                    TreeType = copy.deepcopy(Tree)
                    ModelTijd = TreeType.InorderModel(TreeType.root, Type)
                    if Pruning:
                        TreeType.Pruning(TreeType.root)

                    Runtime = time.time() - StartTime + TreeTime

                    SmallResults = ResultatenZoeker(i, TestSet, TreeType)
                    SmallMetrics = Metrics(SmallResults, Runtime, i)
                    DataLMT = DataLMT.append(SmallResults)
                    ResultsLMT = ResultsLMT.append(SmallMetrics)

                #SVRMT code
                elif Model == 'SVRMT':

                    StartTime = time.time()

                    Type = 'SVR'
                    TreeType = copy.deepcopy(Tree)
                    ModelTijd = TreeType.InorderModel(TreeType.root, Type)
                    if Pruning:
                        TreeType.Pruning(TreeType.root)
                    size = []
                    size = TreeType.TreePrinter(TreeType.root, size)
                    Runtime = time.time() - StartTime + TreeTime

                    SmallResults = ResultatenZoeker(i, TestSet, TreeType)
                    SmallMetrics = Metrics(SmallResults, Runtime, i)
                    SmallMetrics.at[i, 'Size'] = len(size)
                    DataSVRMT = DataSVRMT.append(SmallResults)
                    ResultsSVRMT = ResultsSVRMT.append(SmallMetrics)

            #Write Results to Excel sheets
            ResultsMLR.to_excel(writerResultatenMLR, str(YColumn.split()[0]) + '-' + str(MinSize), index=True)
            ResultsSVR.to_excel(writerResultatenSVR, str(YColumn.split()[0]) + '-' + str(MinSize), index=True)
            ResultsLMT.to_excel(writerResultatenLMT, str(YColumn.split()[0]) + '-' + str(MinSize), index=True)
            ResultsSVRMT.to_excel(writerResultatenSVRMT, str(YColumn.split()[0]) + '-' + str(MinSize), index=True
                )
            ResultsTBR.to_excel(writerResultatenTBR, str(YColumn.split()[0]) + '-' + str(MinSize), index=True)

            DataMLR.to_excel(writerDataMLR, str(YColumn.split()[0]) + str(MinSize), index=True)
            DataSVR.to_excel(writerDataSVR, str(YColumn.split()[0]) + str(MinSize),index=True)
            DataLMT.to_excel(writerDataLMT, str(YColumn.split()[0]) + str(MinSize),index=True)
            DataSVRMT.to_excel(writerDataSVRMT, str(YColumn.split()[0] )+ str(MinSize),index=True)
            DataTBR.to_excel(writerDataTBR, str(YColumn.split()[0] )+ str(MinSize),index=True)


Parameters = pd.Series({'Pruning': Pruning, 'Smooth': Smooth, 'Influence': Influence, 'StandardSplit':Standard, '
    Steps': Steps, 'Split': Split, 'Runs':Runs, 'Seeds': Seeds})

#Write available Input Variables and used parameters to results sheet
AllPredictor = pd.Series(AllColumns)
```

```
for writer in [writerResultatenMLR, writerResultatenSVR, writerResultatenLMT, writerResultatenSVRMT,
    writerResultatenTBR]:
    AllPredictor.to_excel(writer,'Predictors', index=True)
    Parameters.to_excel(writer, 'Parameters', index=True)

#Save Excel Sheets
writerResultatenMLR.save()
writerResultatenSVR.save()
writerResultatenLMT.save()
writerResultatenSVRMT.save()
writerResultatenTBR.save()

writerDataMLR.save()
writerDataSVR.save()
writerDataLMT.save()
writerDataSVRMT.save()
writerDataTBR.save()
```