The application of a Fuzzy Adaptive Learning Control Network in cost estimates for road bridges.

The application of a Fuzzy Adaptive Learning Control Network in cost estimates for road bridges

By

M.S.S. ten Have

in partial fulfilment of the requirements for the degree of

Master of Science
in Construction Management and Engineering

at the Delft University of Technology,
to be defended publicly on February 12, 2021 at 2:00 PM.

Thesis committee:	Prof. dr. ir. A.R.M. Wolfert,	TU Delft
	Dr. ir. R. Binnekamp,	TU Delft
	Dr. E. Demirović	TU Delft
	E. Schulte Fischedick MSc	Witteveen+Bos

An electronic version of this thesis is available at http://repository.tudelft.nl/.
Cover photo: Paul van der Laken

Preface

Dear reader,
My name is Mark ten Have. After obtaining my bachelor's degree in civil engineering I started to attend the master Construction Management and Engineering. In high school, I was always interested in economic issues. During my bachelor's, the interest in economic issues was why I chose a minor in economics, philosophy, technology, and law. The fact that within the master, the possibility to combine economics, cost engineering with technology appealed to me. This interest has led to the search for a graduation topic in this field of interest.

This research is intended for anyone interested in bridges, cost management, or applications of artificial intelligence in civil engineering. The programming process was quite heavy for me. The period of uncertainties surrounding our health due to the COV-19 virus created new challenges. The virus asked and still requires a lot of everyone's adaptability to deal with.

For the supervision during this project, I would like to express my thanks to professor Rogier Wolfert, Ruud Binnekamp, Emir Demirović, and Erik Schulte Fischedick. Besides, I would like to thank my colleagues from the cost engineering group for the period I spent with them at the office of Witteveen+Bos. I also express my appreciation for all the support I have received from my family and friends during this graduation research and study period in Delft.

Mark ten Have
Deventer, February 2021

Summary

In general, costs are an important aspect long before the construction of a civil project starts. Since many projects depend on a budget, it is desirable to obtain an accurate estimate of the costs at an early stage in the development process of the project.
Estimates for civil projects are made by cost estimators who can work on both the client and contractor side of the civil project. Witteveen+Bos is a consultancy company that, among other things, provides advice on costs for its customers. This research project is executed within the Witteveen+Bos company.

The accuracy of a cost estimate depends to a large extent on the amount of available project information on which the cost estimate can be based. Besides, the knowledge and experience of the cost estimator play a role. In the conceptual phase, which is the first phase, of a civil construction project, little cost information is available, and the cost estimate is at that stage of project development less accurate than when the project is almost completed.

In the conceptual phase, the available project cost information is at a maximum 15% of the total information which is available when the project is realized. In the case a cost estimate is made with a conventional method in the conceptual phase of a civil construction project, the Association for the Advancement of Cost Engineering, AACE, argues that the error of a cost estimate compared to the real price can be 15-50% (Bates et al., 2005). The conventional method in the Netherlands, and used by Witteveen+Bos, is the SSK method. This is a format in which project components have been listed according to a fixed pattern after which the total price of the cost estimate is calculated.

The lower accuracy of estimates in the conceptual phase is explainable, diverse obstacles are present in that phase. Serpell (2005) argues that lower accuracy in the conceptual phase is caused by lack of estimating experience, lack of information, and a method that cannot calculate an accurate cost estimate. Besides, the conventional approach for cost estimates is time-consuming (Leśniak & Zima, 2018).

Since all project parties in a construction project commonly rely on conventional methods, the obstacles as presented are still causes of an reduced accuracy of cost estimates (Badra, Badawy, & Attabi). However, Koch (2019) mentioned that machine learning models can improve the performance of cost estimates. Machine learning models calculate cost estimates fast, automated, and accurate (Shin, 2015).

This research was initiated by a practical problem within the company Witteveen+Bos. Witteveen+Bos wants to improve the performance of cost estimates of road bridges in the conceptual phase of a project. This is called an optimization problem (Nehi & Maleki, 2005). In the current situation, the performance is in line with the values of the AACE. This means that the error of cost estimates is between 15-50% and the calculation time takes a couple of hours till days per project.

This problem can be solved by means of a machine learning model. However, if the suitability and benefits of using a specific machine learning model has not been investigated for a specific application, it will not be used for that application (Hong, Wang, Luo, & Zhang, 2020). Although, An, Park, Kang, Cho, and Cho (2007); Elmousalami (2019); Zhou (2018) showed that machine learning models are suitable for solving problems like the presented problem. FALCON is a machine learning model (Abraham, 2004). The FALCON model is already successful applied in the field of cost estimates for construction projects. However, the benefits of this model related to cost estimates for bridge projects are disregarded in literature.

Based on this, the research question follows:
How can FALCON improve the accuracy of cost estimates for road bridge projects in the conceptual phase?

This research aims to improve the cost estimates of bridges in the early phase of bridge projects by use of the FALCON model. Improvement that should be realized in the reduction of the calculation time from hours to minutes. Besides the research aims to reduce the maximum deviation of estimates for road bridge projects to 30%. That is the error of cost estimates compared to the real value 1 step further in the project development process, the budget authorization or control phase, according to the AACE values.

The model FALCON model consists of several steps and is programmed in Python. The predictions of the FALCON model are made based on reference projects, the input data of the FALCON model. The input data of the FALCON model is structured according to the literature. The data is obtained from Witteveen+Bos. Based on the literature, which also includes the Dutch standard for work descriptions RAW, a work break down structure is defined and the cost data per project is divided into nine categories: preparatory work, soil work, foundation structure, substructure, superstructure, pavements, railings, external finishes, and general cost.

Based on the available data set of 39 projects, four categories (foundation structure, substructure, superstructure, and railing) are, on average responsible for 77% of the total construction costs of a road bridge project. The foundation, substructure, superstructure, and railing cost data of 39 reference projects are used as input for the FALCON model. A prediction of the total price is based on the predicted values for each of these four categories. This is equal to the approach of Wang, Bilozerov, Dzeng, Hsiao, and Wang (2017). Since the price levels, years of realization, of the different reference projects differ, they have been adjusted to the same reference year (2020) using the CBS index for bridge construction projects.

The database of the FALCON model consists, besides the cost of the reference projects, of the main cost drivers and properties of each reference project. The FALCON model basically compares the presence of the elements that are the cost drivers and general characteristics of the new project with all projects in the database. Then the best matching project is selected by the FALCON model and the price is predicted for each of the 4 main categories (foundation structure, substructure, superstructure, and railing). A kind of Google form that asks for information about the cost drivers and general characteristics is used to insert the data into the FALCON model on which the FALCON model makes the cost prediction. Cost drivers are the components, e.g. pile type, that contribute significantly to the price of one of the categories.

In the next step, the FALCON model is programmed, and results are generated. Since the four categories are responsible for 77% of the total price. The total predicted construction costs are defined to divide the sum of the four predictions, one per category, by 0.77. The performance of the FALCON model has been tested through n-fold, 10-fold, and 5-fold cross-validation.

In cross-validation, the total used dataset of a model is split into a number of samples, one sample is the test sample, the other samples are used as training set for a model. Then are the results for the test sample calculated by the model. Then the next sample is the test sample and the other samples are the training samples and results are calculated. This process is repeated until each sample has been tested.

The FALCON model is not the only model to solve an estimation problem. Standard, often used, generic models are available to solve estimation problems. Such models, suitable for estimation problems, are multiple linear regression, K nearest neighbors, and decision trees (James, Witten, Hastie, & Tibshirani, 2013; Rockafellar, Uryasev, & Zabarankin, 2008; Strobl, Malley, & Tutz, 2009). These three models have been compared with the FALCON model. Results calculated with these models are based on the same dataset and are also calculated for the same types of cross-validation. The results of all models are shown in table 1.

Table 1: Average error pert type of cross-validation

	KNN	Decision tree	Multiple linear regression	FALCON
N-fold	34%	36%	57%	24%
10-fold	29%	35%	67%	29%
5-fold	41%	36%	68%	30%

Regarding the calculation-time, all models calculate results in minutes. Table 1 showed that the FALCON model calculates the most accurate estimates. The estimation accuracy of the FALCON model does not depend that much on the dataset size, and is, thus, more robust than the KNN model. The models training set size varies for different cross-validation types. Besides, the calculation process of the FALCON model is easy to interpret. This comparison verifies that FALCON generates the best solution to the problem.

The validation of the FALCON model is realized through interviews with cost estimators of Witteveen+Bos. It is checked if the solution to the problem, the FALCON model, meets the requirements of the interviewees. In addition, these interviews check if the research objectives as formulated at the beginning of this study are the same as the requirements of the interviewees. The objective regarding the calculation time of the estimate, determined at the start of this project, has been met during this research and this is in accordance with the interviewees' requirement. In addition, the objective regarding the accuracy of the estimate, determined at the start of this project, has not been met during this research, although the accuracy of the predictions is in accordance with the interviewees' requirement. The average error of all predictions of the models are presented in Table 1. The error of the individual predictions of the FALCON model roughly corresponds to the bandwidth of the AACE (15-50%) for the conceptual phase of a project. Besides, the interviewees are willing to implement the model in practice. This shows that the realized model meets the expectations of the interviewees. In sum, all requirements of the interviewees are met.

Thereafter, the conclusion of this research project follows. FALCON is presented as a model that has the potential to be suitable for cost estimates in the conceptual phase of road bridge projects. FALCON is a model in the field of artificial intelligence. Therefore, a model as FALCON is needs to be programmed to use the model in order to solve the problem as in this research project. The first step to check if FALCON can improve the accuracy of cost estimates for bridge projects is a programmed version of FALCON.

Subsequently the results of this research project are presented. This research shows that FALCON can provide more accurate cost estimates for this problem compared to the generic models. In addition, the FALCON model can make a fast calculation of the total price of a bridge project in the conceptual phase. The improvement is enormous because the time needed for a cost estimate for bridges in the conceptual phase has been reduced from hours to minutes. The accuracy of the estimates is also in accordance with the requirements set by cost estimators in practice, as shown by the interviews with cost estimators from Witteveen+Bos. As mentioned, the FALCON model is not suitable to realize the expected accuracy of cost estimates in the budget authorization or control phase. The accuracy is sufficient for the conceptual phase of a project. Another major advantage is easiness of explaining the FALCON model. That is why the employees of Witteveen+Bos see possibilities and are willing to apply the FALCON model in practice. This contributes to a successful implementation of the FALCON model in practice.

In sum, the FALCON model can calculate a cost estimate, of bridge projects in the conceptual phase, more quickly, and with a comparable accuracy level as with conventional methods that are used today.

Apart from the conclusion, some dependencies and limitations of the model exist. Related to the data, it is important to map if the available data is suitable for the intended research strategy. Besides, data pre-preprocessing can be a time-consuming process. This must be considered. It is unknown whether this model provides the best solution to the problem, therefore, more research on the subject is necessary.

Besides the accuracy and processing of the data, the comparability of projects is an issue. For the size of the dataset, it may be that the variation is too large. In case the dataset is larger or has less variation, the accuracy of the predictions could increase. A larger dataset is recommended in future research. The data of the model is structured using the Dutch standard, adjusting the FALCON model to an international context requires some effort. In future research, it is therefore recommended to use international standards for the implementation of models such as FALCON in an international context.

Contents

1 Introduction

This chapter introduces the project, which is executed within the Witteveen+Bos company, and the research problem. Paragraph 1.1 provides the research context. This section is followed by the problem statement in paragraph 1.2. The main question of this study follows from the problem statement, which is described in section 1.3. Subsequently, in section 1.4 the problem type is described with the corresponding approach to such problems. The final paragraph, 1.5, presents a reading guide for this report.

1.1 Research context

This research takes place in the field of cost engineering and this paragraph introduces that field of interest.

In general, costs are an important aspect long before the construction of a civil project starts. Cost is one of the main measures of the success of a civil project. Therefore, related to this main measure of success, cost estimates are important for several reasons. These reasons are budget definition, loan application in the case a project needs funding, and estimating the likely cost of a loan (Ahiaga-Dagbui & Smith, 2014). Besides, the accuracy of cost estimates is crucial in the determination process whether or not a project is undertaken or infeasible (Lim, Nepal, Skitmore, & Xiong, 2016).

However, a cost estimate must be accurate to use the estimate for the above-mentioned arguments. The experience of cost estimators, the availability of reference projects, the completeness of the new project's information, and the estimating method are factors that are related to the accuracy of a cost estimate (Hatamleh, Hiyassat, Sweis, & Sweis, 2018). The completeness of the new information is a factor that differs per project life cycle phase. Therefore, the accuracy of an estimate differs per project life cycle phase. The project life cycle phases are the conceptual, design, realization, and operation phase. Figure 1 shows each phase and accuracy of cost estimates in that phase (Burke, 1999).

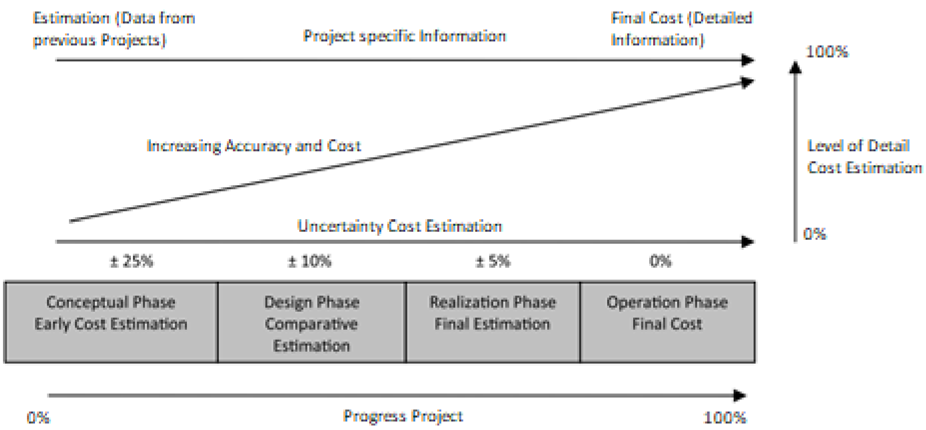

Figure 1 Accuracy of the cost estimate and progress of the project over time.

Besides, AACE International (Association for the Advancement of Cost Engineering) defines an accuracy range for a specific purpose. According to AACE International, the expected accuracy is 15-50% in case the purpose of the estimate is a standard study or feasibility study. This largely corresponds to the conceptual phase as described above in figure 1 (Bates et al., 2005).

From the numbers of the AACE and Burke, it is clear that the estimates are less accurate in the conceptual phase than in the operation phase. The reduced accuracy is explainable. In the conceptual phase in the development of civil construction projects, diverse obstacles are present that result in a reduced accuracy of cost estimates. Serpell (2005) argues that a reduced accuracy in the conceptual phase is caused by lack of estimating experience, information, and an method that is able to calculate an accurate cost estimates in the conceptual phase. Besides, the conventional approach of cost estimating is a time-consuming process. In the conventional approach estimates are made separately for direct cost (equipment, labour and materials), indirect cost, and profit (Leśniak & Zima, 2018).

Since all project parties in a construction project commonly rely on conventional methods, the obstacles as presented are still causes of an reduced accuracy of cost estimates (Badra et al.). However, Koch (2019) mentioned that machine learning can improve the performance of cost estimates. Machine learning models calculate cost estimates fast, automated, and accurate (Shin, 2015).

1.2 Problem statement

This research was initiated by a practical problem within the company Witteveen+Bos. This practical problem relates to the obstacles as presented in paragraph 1.1. Witteveen+Bos wants to improve the performance of cost estimates of road bridges in the conceptual phase of a project. The objective is to come up with a more accurate and fast calculated cost estimate in the conceptual phase of road bridge projects.

In the current situation, a conventional approach of cost estimating is used. This conventional approach in the Netherlands, and used by Witteveen+Bos, is the SSK method. This is a format in which project components have been listed according to a fixed pattern, then the cost estimate of the total project is the sum of the estimates of all components.

The performance of that approach is in line with the values of the AACE as mentioned in paragraph 1.1 which means that the accuracy is between 15-50% and the calculation time takes a couple of hours till days per project. This problem ties in well with the motive in construction management. The motive in construction management is to strive for optimal project performance related to time, cost and quality (Koch, 2019).

The former section 1.1 already described the importance of cost estimates, and it describes that machine learning models can improve the performance of cost estimates. FALCON is a machine learning model which is already applied in the field of cost estimates for construction projects (Abraham, 2004). However, this model is not applied for cost estimates of road bridges in the conceptual phase.

1.3 Research question

Based on the context described earlier in this chapter, the research question for this project is formulated.

How can FALCON improve the accuracy of cost estimates for road bridge projects in the conceptual phase?

1.4 Problem type

This project starts with the strategy to find an answer to the main question of this master thesis. This research is identified as a problem in the optimization of cost estimates for road bridges, therefore a new system needs to be developed. This problem is therefore a development problem.

This research project is executed according to the scheme for development problems which is developed by Roozenburg and Eekels (1995), and presented in figure 2. As can be seen in figure 2, there are six steps that need to be taken after determining the problem in order to arrive at the solution, a new engineering system.

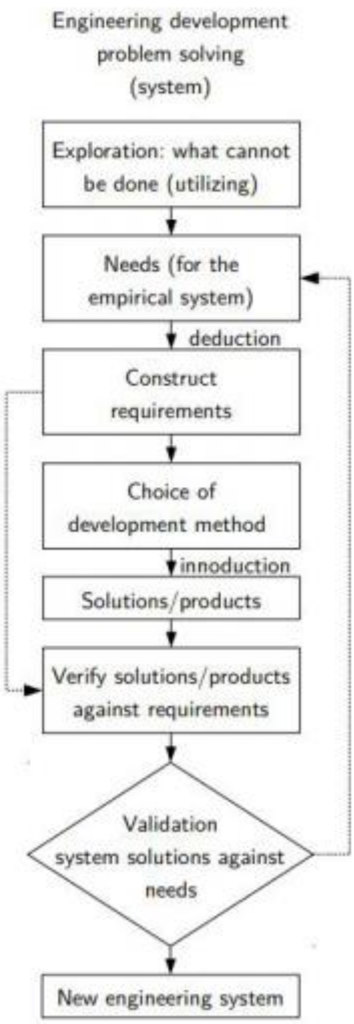

Figure 2: Flow chart development new engineering system (Roozenburg & Eekels, 1995)

As seen from top to bottom in figure 2:
The first step is the Exploration. This step represents the problem for which a new engineering system is necessary.
Second, to solve the problem, the new engineering system must meet several requirements. These requirements are the 'needs'.
Third, the construct requirements are deduced from the needs. The 'construct requirements' are the requirements that are set for a model that may provide a solution that may meet the 'needs'. The construct requirements are not the same as the 'needs'.
Fourth, based on the construct requirements, it is an option that more than one method can be used to find a solution for the problem. Thus, a choice for one or more development methods is made.
Fifth, the method is used to come up with solutions for the problem.
The sixth step is the verification of the solutions with the 'construct requirements'. If the requirements are not met, another method must be used to find a solution that meets the 'construct requirements'.
The seventh and final step is the validation. In this step of the scheme, it is checked if the solution fulfils the needs.

Paragraph 1.1 presented that the performance of cost estimates in the conceptual phase of construction projects is influenced negatively by some obstacles. Witteveen+Bos deals with a problem that relates to the obstacles. The problem describes that it is not possible to calculate an accurate cost estimate for road bridges in the conceptual phase, this is the 'Exploration' as described in figure 2.

1.5 Reading guide

Paragraph 1.4 presented the research strategy. Based on this strategy, the research of this project is structured.

This chapter, the introduction, presented already the Exploration. Chapter 2 presents a context analysis of the problem, the 'needs' are presented in this chapter. Besides, Chapter 2 presents that FALCON is not the only model that may be suitable to solve this research problem. Chapter 3 describes the models that are suitable to solve this research problem.

Thereafter, chapter 4 shows the method which is used to calculate the results. Besides the results are verified with the construct requirements. Thereafter the validation phase is described in chapter 5. The validation step is followed by the concluding chapter, chapter 6, of this research project, not shown in figure 2. In this chapter, the conclusion, discussion and recommendations are presented.

This research report is structured according to the strategy of figure 2. Figure 3 shows a visualization of the research strategy steps per chapter.

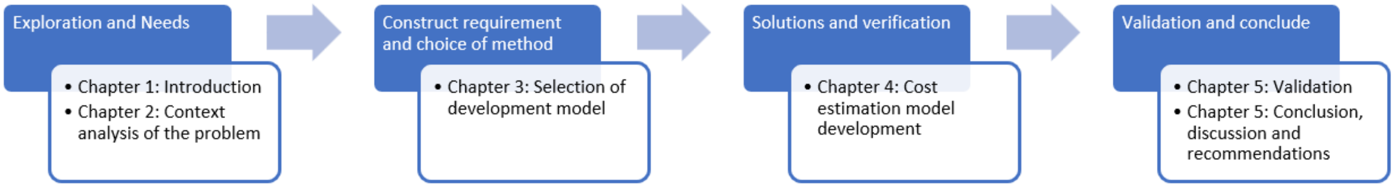

Figure 3: Structure of the research, an overview

2 Context analysis of the problem

In the first chapter, an introduction of the research project is given. This chapter elaborates on the context of the research project. The first paragraph elaborates on the content related context of the research problem. The second paragraph describes the research related context. One of the aspects of the second paragraph is the description of the ‘needs’, the second step of the research strategy as presented in chapter 1.

2.1 Content related context

The first chapter presented machine learning models as solution for the obstacles in practice. Machine learning models are diverse and are applied in different fields of interest. Section 2.1.1 shows that machine learning models are a logical choice to solve this type of problem, making an accurate cost estimate.

The literature shows that there are possibilities for various machine learning models in the field of cost estimates. The literature also shows through recommendations that the neuro-fuzzy machine learning models can be further investigated. Section 2.1.2 describes this in more detail.

Section 2.1.3 elaborates on the performance of FALCON in civil engineering projects. Promising results are presented.

The context shows that it is likely that the FALCON model could be a suitable solution to the problem outlined in chapter 1. Despite the recommendations, and previous results of the FALCON model in other areas, this model may not provide the best results. will provide. Section 2.1.4 shows other models that can be suitable to solve this research problem.

2.1.1 Introduction different types of machine learning

Machine learning is one of the fastest-growing computer science areas, with far-reaching applications (Shalev-Shwartz & Ben-David, 2014). There are different types of machine learning systems. Most of them are learning proactively, the model actions as determined on beforehand. However, instance-based learning systems, also called case-based reasoning systems are reactive learners (Reich, 1997). Instance-based learning systems will do predictions based on reference data.

Machine learning is divided into two main categories, supervised learning and unsupervised learning. Both categories have two subcategories presented in figure 4. In the case of supervised learning, prior knowledge is available what the output should be. The goal of supervised learning is to come with the best approximation between input and desired output. This goal ties in well with this research problem in which the goal is to make an accurate cost estimate based on reference data. A cost estimate is a continuous number, not a classification. The machine learning model type is, therefore, a regression model.
In the case of unsupervised learning, no labelled output is available. The main goal for this type of learning is finding a natural structure within the data (Brownlee, 2016).

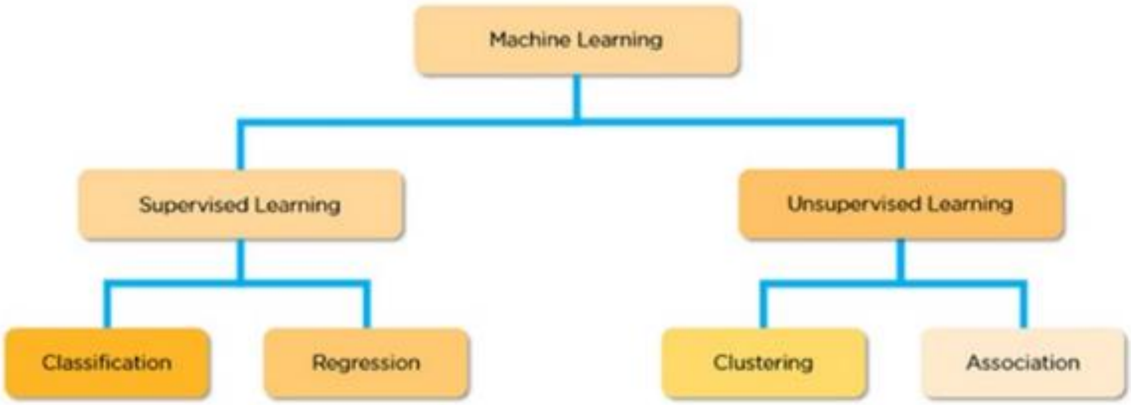

Figure 4 Machine learning algorithms (Brownlee, 2016)

2.1.2 Appropriateness of diverse machine learning models for cost estimates

Referring to literature, it is evident that several machine learning models are able to improve cost estimates. Zhou (2018), presents the improvement of cost estimates in the early phase of a project using the machine learning model ANFIS. An et al. (2007) showed the applicability of the machine learning model SVM (Support Vector Machine) for the improvement of conceptual cost estimates of construction projects. Elmousalami (2019) investigated the traditional fuzzy machine learning model, and the modified variant of this model with a genetic algorithm, for cost estimates in canal improvement projects in Egypt.

Additionally, the application of other machine learning models for increasing performance could show promising results and they should be further investigated according to Elmousalami (2019); Koch (2019). In that perspective, Elmousalami (2019) recommended to investigate neuro-fuzzy models in which MF Membership Functions are included. A membership function represents with a gradual transition function if a certain parameter 'x' is part of a set or not, and represented by a value between 0 and 1 (Idrus, Nuruddin, & Rohman, 2011; Koch, 2019). For example, a triangular membership function can represent the degree of truth to which a point with colour 'x' is equal to apple green.

Neuro-fuzzy models are hybrid models that can be evolved from fuzzy logic models (Elmousalami, 2020). Neuro-fuzzy models are already used for cost estimates of construction projects (Cheng, Tsai, & Hsieh, 2009), and able to solve regression problems (Shihabudheen & Pillai, 2017). Fuzzy logic is an extension of Boolean logic. Boolean logic uses the operators AND, OR, and NOT. Fuzzy logic uses all values from 0 to 1. Besides false and true, a value in between is thus possible (Zadeh, Yager, Ovchinnikov, Tong, & Nguyen, 1987). FALCON is a neuro-fuzzy model (Abraham, 2004), and this model is not applied for cost estimates of road bridges in the conceptual phase.

2.1.3 Applications FALCON

Various research projects have been carried out on the benefits of FALCON in construction projects. However, the benefits related to cost estimates for bridge projects are disregarded in literature. A hybrid model was applied for piping systems by Hsiao, Wang, Wang, Wen, and Yu (2012). Hsiao et al. (2012) used FALCON (Fuzzy Adaptive Learning Control Network) standalone, and FALCON together with fmGA (fast messy Genetic Algorithm) to enhance cost estimate accuracy. A genetic algorithm applies the process of natural section to other field in order to produce a better result based on references. The research of Hsiao et al. (2012) showed that the FALCON model has an accuracy of 84% and is more accurate than the conventional cost estimation method for construction projects. For concrete building projects, an accuracy of 91% is reached, using a small dataset of 46 projects (Wang et al., 2017).

2.1.4 Standard models for estimation problems

As mentioned in section 2.1.3, the results of the FALCON model are very promising in various fields of civil engineering. It is therefore also plausible that the results for cost estimates of bridges using the FALCON model are better than standard, often used, generic models that can solve estimation problems. Such models, suitable for estimation problems, are multiple linear regression, K nearest neighbors, and decision trees (James et al., 2013; Rockafellar et al., 2008; Strobl et al., 2009).

2.2 Research related context

The second part of this chapter is the context related to the research process itself. First, section 2.2.1 describes the need to solve the research problem as presented. Thereafter, in section 2.2.2, the ‘needs’ are defined. The research is executed within a certain scope, this scope is described in section 2.2.3. During the research, two types of knowledge are used. These types of knowledge are described in section 2.2.4.

2.2.1 Need for solving this research problem

Chapter 1 and paragraph 2.1 showed that there is a practical problem that may be solved using the FALCON model. As mentioned, Witteveen+Bos wants to increase the accuracy of cost estimates in the conceptual phase of bridge projects. One of the reasons for this is that the budgets of Witteveen+Bos customers are based on Witteveen+Bos’ cost estimates.

However, before the FALCON model can be applied in practice, it must be proven that the FALCON model can solve the research problem. If the suitability and benefits of using a specific machine learning model has not been investigated for a specific application, it will not be used for that application (Hong et al., 2020).

2.2.2 Definition of the ‘needs’

As mentioned earlier in this chapter, this research further explores the applications of neuro-fuzzy models in the field of cost estimates for construction projects. More specific, this research has a twofold purpose, on the one hand, this research solves an optimization problem in the field of cost estimates, on the other hand, this research examines a model application whose need for this research is apparent from a recommendation in previous research.

The main research objectives are:

- To increase the performance of cost estimates for road bridges in the conceptual phase based on the major targets:
    - Accuracy
    - Calculation time

This research objective shows two ‘needs’ according to the research strategy for a new engineering system as presented in paragraph 1.4. The two ‘needs’, one regarding accuracy and one regarding the calculation time are the requirements for the new engineering system. When the system is realized, it must be validated if the system meets the ‘needs’ regarding the accuracy and the calculation time.

The ‘needs’ of the new engineering system are to realize significant improvements in the performance of cost estimates in order to achieve a successful outcome that leads to adaptation of current working methods.

The ‘needs’ are formulated below:

Calculation time:

The required calculation time must be reduced from hours to minutes. This enables Witteveen+Bos to respond quickly to customer questions. This contributes to customer satisfaction.

Accuracy:

For Witteveen+Bos, a major step has been taken if the accuracy is improved one level based on the AACE values. This means that the expected deviation on the predicted price with the model may be a maximum of 30%. A value that normally is set for budget authorization or control (Bates et al., 2005).

2.2.3 Scope of the research

The scope of this project is already largely disclosed in the main question.

The project focuses on the conceptual phase of road bridge projects as already became clear in the former sections. More specifically, road bridge projects which are not part of the main road structure. The choice of these bridge types was made because the data, which will be described in more detail later, consists of such projects.

The scope of this project is aimed at cost estimators who are active in the field of road bridge construction projects. Since this research is executed within the Witteveen+Bos company, these cost engineers participate in this research project they give their experts’ opinions in the model evaluation phase.

2.2.4 Knowledge

The knowledge used in this project is gathered using two methods: literature studies and expert interviews are executed.

2.2.4.1 Literature studies

Previous research on a topic is presented in an extensive summary. The literature studies explain the used models in more detail. The literature study aims to understand the theory of the models, and based on this theory, the models are used to calculate cost estimates for road bridge projects. The literature study is part of the first phase of the research project.

2.2.4.2 Expert interviews

The main reason for doing expert interviews is to validate if the solution, a new engineering system, meets the ‘needs’. Besides, the interviews aim to gather feedback regarding the applicability of the model. Therefore, seven interviews with cost engineers of the Witteveen+Bos will be performed.

3 Selection of development model

In chapter 2, the context analysis showed that FALCON is not the only model which can be used to solve the research problem. There are therefore several options within all solutions to meet the 'needs'. The construct requirement is: finding the best performing cost estimation model related to accuracy and calculation time. The ‘choice of development method’ step, as presented in paragraph 1.4, consists of the choice for the investigation of four models in order to find the best performing cost estimation model.

These four models, FALCON, multiple linear regression, decision trees, and K nearest neighbors, are programmed in Python. Using the programmed versions, cost estimates for road bridges are made. To be able to program the models and to be able to assess the results, detailed literature of these models is required. The detailed literature is outlined in this chapter. That means that almost all the necessary literature for this research project is brought together in this chapter.

First all 4 models are presented. Paragraph 3.1 presents the FALCON model, paragraph 3.2 presents the multiple linear regression method, followed by the decision trees in paragraph 3.3. The last model, the K nearest neighbors model, is presented in paragraph 3.4. Paragraph 3.5 presents, based on the described theory of the four models, the pros and cons of each of the four models. A first insight into the expected performance of the models is provided in this section.

Second, the models need structured cost data as input. The second part describes, in paragraph 3.6, how cost data of reference projects can be structured to make this data suitable for making cost estimates of new road bridge projects. The final paragraph, paragraph 3.7, describes the theory about the calculation of results with computer models.

3.1 FALCON model

This section describes the FALCON model in more detail, based on various sources. FALCON as a model is represented by a five-layered structure as represented in figure 5 (Abraham, 2004). The FALCON model does not have the ability to change this structure of the network dynamically (C.-T. Lin, Lin, & Lee, 1995).

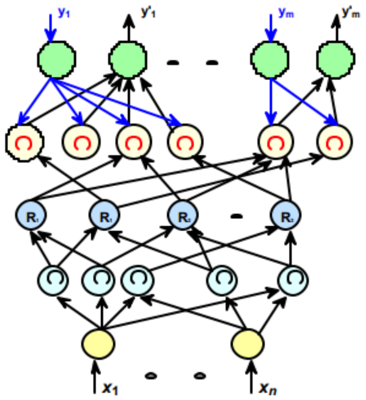

Figure 5 Architecture of FALCON

In the first layer, as seen from bottom to top, the input's fuzzification takes place. Fuzzification means the decomposing process from a normal input set with values to one or more fuzzy sets. Layer 1 just transmits the input values directly to the next layer (C.-J. Lin & Lin, 1997). Each node in this layer in the system represents a simple membership function (MF). A membership function represents, with a gradual transition function, if a certain parameter ‘x’ is part of a set or not, and represented by a value between 0 and 1 (Idrus et al., 2011; Koch, 2019). For example, a triangular membership function can represent the degree of truth to which a point with colour ‘x’ is equal to apple green. However, not all nodes do necessarily have the same membership function.

A membership function processes all input into membership values, the output of a membership function. As mentioned earlier, these values are between 0 and 1, and the membership function represents a graph. One of the suitable membership functions for the FALCON model is the Gaussian function (Nikam, Nikumbh, & Kulkarni, 2012). Other options are the triangular and trapezoidal function (Rodríguez, Falcón, Varela, & García, 2008). It is not established which function is the most suitable. For example, it is possible to visualize the distribution of the data. This can be used to define the most suitable function. Trial and error is another option (Brownlee, 2016).

A node in the network can be represented by a scheme, which is presented in figure 6. It is possible to simplify a neuron as a point with different inputs. The sum of the inputs is always one and means that the input must be weighted if a node has more than one input value.

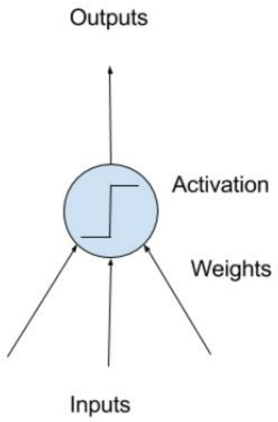

Figure 6 Schematized node of a network (Brownlee, 2016)

The second layer, as seen from bottom to top is one of the hidden layers in the network. A hidden layer has input that is formed inside the network. In the second layer, preconditions are defined for the fuzzy rules in layer 3. A fuzzy rule looks like this: 'If x is A, then y is B.' A and B are linguistic values. It assumes that these values are derived from statistical research. If the rule consists of two parts, then the first part,' x is A,' should be called the antecedent or premise, and the second part, ' y is B,' should be called the consequent or conclusion. In layer 2, the antecedent or premise is defined. The nodes in this layer are called input-term nodes (C.-J. Lin & Lin, 1997).

If the number of preconditions based on the input variables increases, the number of the fuzzy rules grows combinatorically. Then, the fuzzy rules create a finer space partitioning, and may need more training samples (C.-T. Lin et al., 1995).

The third layer consists of rule nodes. Each of the nodes in this third layer is representing a fuzzy logic rule. These rules are not predefined. There are 'n' input terms that fed the node of layer three. The number of rules is equal to the number of terms of the input variable (C.-J. Lin & Lin, 1997).

In layer 4, the nodes are called output-term nodes, the output of the fuzzy rules. Each node has two functions, a down-up, and an up-down mode. Figure 5 is also showing this. In the down-up direction, the layer performs as an OR operation on the fired (activated) rule nodes that have the same consequent. The OR operation means that the nodes process a max function. E.g., f(x) = max(x1, x2, x3). There are several possible solutions available in the database. This layer, therefore, examines which solution from the database of references best suits to all answers of the fuzzy rules. In the opposite direction, the nodes function precisely as the nodes in layer 2 (C.-J. Lin & Lin, 1997). Layer 4 helps in deriving the consequents of the fuzzy rules.

In the final layer, layer 5, is also performing a down-up and up-down function. The up-down function transmits training data into the network. This information flows from the output to the input of the network. The concept is called 'backpropagation.' This process is useful because it improves the network's accuracy (Brownlee, 2016). For this function, layer five is working as layer 1 (C.-J. Lin & Lin, 1997). In the down-up function, defuzzification takes place. Defuzzification takes place using the 'centroid method.' This method is searching for the best outcome by clustering the outcomes that are the most similar to the desired goal (Brownlee, 2016).

3.2 Multiple linear regression

A simple method to find an answer to this problem would be the multiple regression model. This model is one of the standard solutions for estimation problems (Rockafellar et al., 2008). In such a model, more than one regressor variable is available.

For example: Y= B0 +B1*X1 +B2*X2 + E.

In this function, X1 could be the bridge deck in square meters, and X2 could be the number of supports. This example is a multiple linear regression model with two regressor variables. The function calls multiple linear regression because it is a linear function of the unknown parameters B0, B1, and B2. E is the error term.

To define our total price (Y), more than two regressor variables are used: Y= B0 +B1*X1 +B2*X2 + … + Bk*Xk + E. The 'B' parameters are called the regression coefficients. These parameters represent the expected change in Y (total price) per unit change when all other parameters are still the same.

The method of least squares is used to define the regression coefficients. This simple estimation method requires an extensive historical data base to accurately determine the regression coefficients (Mahamid, 2011).

3.3 Decision tree regression

This section describes the decision tree regression model in more detail, based on various sources. Classification trees and regression trees are a simple regression approach. The approximation of bridge prices can be classified as a regression problem according to figure 4 in chapter 2. In this method, the feature space is the space spanned by all predictor variables. This feature space is recursively partitioned into a set of rectangular areas. Figure 7 shows an example of predicting the intention of smoking by adolescents depending on a couple of factors.

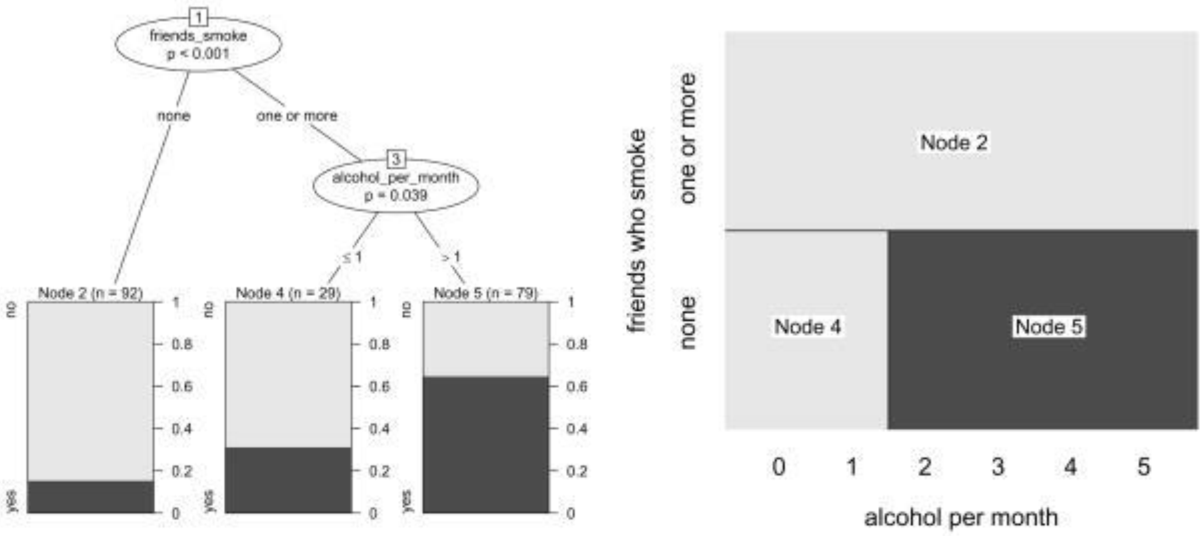

Figure 7: Example Decision tree regression (Strobl et al., 2009)

Each partition grouped observations with similar response values (Strobl et al., 2009). After these partitions, the decision tree regressor predicts a value.

The representation can be as a rectangular partition of the feature space or as a tree. These rectangular partitions are one of the features that are different from normal regression. Normal linear regression combines the information linearly. In this case, all possible combinations are generated by recursive splitting. In recursive splitting, the data is split into sub-populations based on dichotomous independent variables. Multiple splits in the same variable can be derived. In short, this means that the model generates nonlinear and non-monotone rules. Remarkable for these rules is that the conclusion does not necessarily follow from the premises. Necessary inference is not possible because the available knowledge may be incomplete. In non-monotone-logics, a conclusion is valid as no new knowledge becomes known that invalidates the conclusion. These rules are not specified in advance. They are determined in a data-driven way.

To do these splits, the model uses impurity reduction. Impurity is a factor that made clear how often a randomly chosen element from a dataset is labelled incorrectly in case it is labelled randomly according to the distribution of the dataset. The impurity of a node should thus be defined. A node that has no impurity has no variability in the dependent variable. For example, all values are equal to zero or one. In case the values in a node are equal to zero or one, then the highest value of impurity is equal to 0.5. The splitting criterion selects the split that has the largest difference between the parent node's impurity and a weighted average of the impurity of the two child nodes (Lemon, Roy, Clark, Friedmann, & Rakowski, 2003). In figure 7, ‘friends smoke’ is the parent node, ‘one’ and ‘one or more’ represent the child nodes.

This splitting process stops at some point. The model reaches a stop condition in that case. There are some different stop conditions: all nodes are pure (for example; only zeros or ones are present), the model reaches a given threshold for the minimum number of observations left in a node, or a given threshold for the minimum change in the impurity measure is not succeeded any more by any variable (Strobl et al., 2009). Due to these rules, the model misses probably some information.

In the end, the model predicts a response value in each terminal node of the tree. From all observations in this node, the model derives the average response value. A classification tree derives the most frequent response class from all observations (Strobl et al., 2009). The response value becomes more accurate as the dataset grows (Oates & Jensen, 1997).

3.4 K nearest neighbors

This section describes the K nearest neighbors model in more detail, based on various sources. K nearest neighbors is a simple classification method. This classification method can be applied when little information is available about the distribution of the data. This theory is developed as a solution for the case in which it is difficult to determine the probability densities (Peterson, 2009).
In this project's case, the number of parameters makes it difficult to plot a distribution function in which they are all included and give a true reflection of the dataset.

The k nearest neighbors' classifier (KNN classifier) defines ‘K’ points in the data set that were the closest to test point x0, the test project, represented by N0. The conditional probability is calculated for class j as the fraction of point in N0 whose response value or values are equal to j (James et al., 2013). This results in equation 3.

Equation 1: KNN conditional probability

$$\Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in \mathcal{N}_0} I(y_i = j).$$

The conditional probability equation is visualized in figure 8. In case k = 3, the red star is classified to be of class B, in case k = 6, the majority changed to class A and the red star is classified to be of class A. The sensitivity to the value of k in case of small training samples with existing outliers can cause less accurate results (Gou et al., 2019).

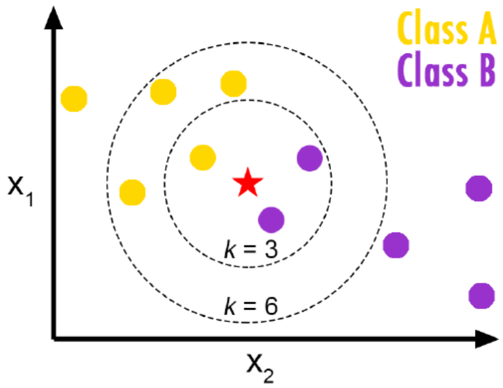

Figure 8: KNN conditional probability (Lamein, 2016)

A classification method is not very useful in case prices should be estimated. In case the cost of all bridge projects is divided into certain classes, only the 'cost class' can be established in that case.

Using a variant of KNN classification is an option in case a continuous variable should be estimated. This variant is KNN regression. KNN first identifies the K training observations that are closest to x0. N0 represents them. Taking the average of all training responses gives the final result f(x0). (James et al., 2013).

Equation 4 shows the KNN formula.

Equation 2: KNN regression

$$\hat{f}(x_0) = \frac{1}{K} \sum_{x_i \in \mathcal{N}_0} y_i.$$

3.5 Evaluation theory cost models

Paragraph 3.1, 3.2, 3.3 and 3.4 described the four different models that are used in this research project. Table 2 listed the characteristics of each model.

Table 2: Characteristics cost estimation models

KNN	Decision tree	Multiple linear regression	FALCON
Works well when less information is known about the data.	Creates partitions with similar values	Assumes a linear relationship in the data.	Best suitable membership function is not defined
Result is average of ‘k’ nearest neighbors.	Response does not necessarily follow from the premises.	Extensive historical dataset is necessary for an accurate determination of the regression coefficients.	Outcome based on the most similar references
Outliers in data can cause less accurate predictions.	Response is average of datapoints in the leaf of a tree.	Simple method, less complex	More complex system with 5 layers.
	Rules are not specified in advance; they are based on the available data.		Fuzzy rules do not need to be predefined.
	Predictions more accurate as the dataset grows.		The structure of the model does not change dynamically
			The more detailed the result must be determined; the more reference data is required.
			Reference concrete buildings shows that the model is accurate for a small dataset. (paragraph 2.1)

From table 2, several advantages and disadvantages are derived regarding this research

A main advantage for KNN is the fact that this model works well in case less information is known from the data. However, this also has a disadvantage, the dataset that is used is not very large, so possible outliers in the data can influence the results generated with the model.

A main advantage for the decision tree is that this method is less sensitive to outliers since this method calculates the result as average of the projects in a leaf of a tree. A disadvantage is that the outcome of the model does not necessarily follow from the data. The decision tree does not split the data on the basis of the same condition every time. This means that the data can be split, for example, first based on the number of supports of a bridge, and then based on the surface of the bridge deck, and then on material. This order can also be completely different. There are therefore several outcomes possible for a test case. Another disadvantage is that the results with this method can be better if a larger dataset is available.

The biggest advantage for the multiple linear regression is the simplicity of the model. The model represents one quite simple formula. The disadvantage for this model is the need for a large dataset to determine the coefficients accurately. Besides, this model assumes a linear relationship between the datapoints, if this is not the case, the predictions may be less accurate.

An advantage for FALCON is the reference that the model is suitable for small datasets. Another advantage is that the fuzzy rules that the FALCON model uses do not need to be predefined. Extensive knowledge about the data is thus not necessary. Another advantage is that the model does not change dynamically. This keeps the model clearer and makes it easier to understand in different situations. I addition, disadvantages for this model exist. The model needs more reference data in for detailed predictions. The model is more complex, complexity can be a disadvantage if the model is used for a new application.

In sum, each model has advantages and disadvantages, so that it is not yet possible to determine in advance which model will have the best results independent on the used data. Although it seems likely that the predictions of the multiple linear regression model will be worse because this model requires a large dataset.

3.6 Input data

The models described in paragraph 3.1, paragraph 3.2, paragraph 3.3, and paragraph 3.4, use structured cost data from different projects. Various sources are available in the literature to structure the currently relatively unstructured cost data of projects. This section will gather the data applicable to this research project. Thus, a way will be presented how the cost data of the bridge projects can be structured.

The projects are compared to each other on a certain level of detail by the models. That level of detail is related to the amount of information available in the conceptual phase of a bridge project.

The first general step to make the data of different projects from the past comparable is that the data must have the same price level. This is described in section 3.6.1.

To reduce the variation between the cost data of the projects, it is possible to divide bridges by type, which is described in section 3.6.2.

As mentioned earlier, the bridge projects are compared to each other by the models at a certain level of detail. First, the level of detail is described in section 3.6.3. A bridge can be divided into parts based on the level of detail. This is described in section 3.6.4.

The conventional method for making cost estimates is the SSK method. This method is used by both clients and contractors. The available data from reference projects is also structured in this way. Therefore, the SSK method is described in section 3.6.5.

3.6.1 Index

For the cost, data swaps are necessary regarding the index of prices. Swapping the data is one of the steps to make data comparable. For example, if the model takes a project that is executed several years ago and this price in the database is not corrected with indices, it is not possible to buy at this moment the same product for that price because the prices have increased since the execution of the reference project.
In the case of road bridge projects, an index table regarding road bridges is the most optimal. The index numbers have the best fit in that case. Especially considering this is the type of project used in this study. In the Netherlands, an index for bridges is available. This index is created by Calcsoft BV and available by use of a license of Witteveen+Bos. Calcsoft BV presents index numbers from 2008 until now.

Calcsoft does not present one single index number. They present an index number for the construction cost index and the tender index, (in Dutch: 'bouwkostenindex' and the 'aanbestedingsindex'). The construction cost index visualizes the development of the actual construction costs over time. The tender index visualizes the development of the bids made by contractors on projects. The difference between these indexes does suggest the profits within the companies.

More than one index is available for bridges. The governmental organization CBS (Centraal Bureau voor de Statistiek) presents index numbers for bridges for decades. This creates an opportunity to implement more project data. This research project also uses cost data of projects that were carried out in the period before the year 2008. Only the annual indexes of CBS are therefore suitable for this research project.

Appendix I presents the accompanying index tables and graphs of both indexes.

3.6.2 Categorization on bridge type

To reduce the variation between the cost data of the projects, it is possible to divide bridges by type. The used bridge projects in the database of Witteveen+Bos are all bridge projects in municipalities in the Netherlands, which have not been realized as part of a provincial road or road from highways' main structure. These bridges are part of the secondary road network.

The municipality Wormerland has made a general division of all their bridges, as shown below (Van der Linde, Modders, & Van Dissel, 2017).

1. Movable road bridges
2. Concrete road bridges
3. Concrete bicycle/pedestrian bridges
4. Wooden road bridges
5. Wooden bicycle/pedestrian bridges
6. Steel road bridges
7. Steel bicycle/pedestrian bridges
8. Plastic bridges

The division of the municipality Wormerland is a general division.

3.6.3 Level of detail

On a project level, a division in different elements must be made. Much detailed information makes it simpler to make an accurate cost estimate of a bridge. A break down structure on four levels is formulated by Du Bois, Fletcher, and Danks (2017), and the text below shows this structure.

Level 1: Elemental level
Level 2: Heading level
Level 3: Item level
Level 4: Rate Build up level

Level 4 contains the least detail within this structure. Level 3 already describes a bit more about the details. For example, the bridge deck consists of a deck and a railing. Even more detail compared to level 3 is given by level 2. In the case of the bridge deck, prefab beams are one of the headings. The lowest level in this structure is level 1. That case concerns a prefabricated beam including mounting material.

This break down structure above shows that the difference per level is detail based. When making a cost estimate, it is important to know the level of detail of the available information. This also determines the level of detail at which a cost estimate can be made.

Regarding level 2, heading level, all cost drivers must be clear after writing down all headings. Cost drivers are parameters that have a direct impact on cost. These cost drivers can be related to scope, complexity, performance. They can be identified through regression analysis to define the impact trend with the cost. Sometimes a proxy variable is available and used for a set of known variables. It is also possible to define cost drivers using the available data (Dechoretz, 2011). The latter option is used in this project, and in this research means available data obtained from historical projects.

Wang et al. (2017) had used the component ratios method to focus on the major costs of a project before the FALCON model was applied. The component ratio method is a type of work break down structure. This approach resulted in ten cost divisions: foundation, structure, external finishes, internal finishes, windows, MEP (Mechanical / Electrical / Plumbing), elevator, temporary facilities, landscaping, and markup, and finally, four divisions, cost drivers, were responsible for 71,02% of the cost of a project and used as input for the model.

In this project, the component ratio method proposed in the paragraph above is used as well. Since a bridge is another project type, the components are different from those chosen by Wang et al. (2017).

3.6.4 Work break down structure

If a level or detail has been determined, a project can be divided into different components using a work breakdown structure. This structure can be different for each type of construction, dependent on the level of detail.
In the previous section different levels of detail have been described. For bridging projects, applications can also be found in the literature in which a detail level has been chosen for the work break down structure of a project. This section describes several studies.

Concerning steel bridge projects, Nugroho and Latief (2018) suggested a work breakdown with eight divisions:
1. Preparatory work
2. Drainage work
3. Land works
4. Pavement widening and shoulder widening works
5. Asphalt pavement
6. Structural work
7. Toll service facility
8. Return of minor conditions

It is also possible to define a project in ten divisions. Latief, Nurdiani, and Supriadi (2019) suggest this division for steel bridge projects. The ten divisions are:
1. General
2. Drainage
3. Earth Works
4. Pavement widening and shoulders
5. Concrete powder and concrete cement
6. Asphalt pavement
7. Structure
8. Restoration condition
9. Daily work
10. Daily maintenance

The provided divisions above are only for steel bridges. Kurnia, Latief, and Riantini (2018) provide a division for precast concrete bridges. Besides, that research showed that most packages as listed below were the same for different types of bridges: steel bridges, cable-stayed bridges, flyovers, precast bridges, and roads. Only the structure work is different for all types.

1. Preliminary work
2. Drainage work
3. Soil work
4. The spread of pavement and shoulders
5. Coated hardening and concrete pavement
6. Asphalt pavement
7. Structure work
8. Toll service facility
9. Recondition of minor work

However, the lists described above are not fixed divisions. A project manager is responsible for the creation of a work breakdown structure according to Burghate (2018). These work packages defined by the project manager should be: definable, manageable, estimable, independent, integrative, measurable, and adaptable (Rev, 2003).

In order to be able to define the cost drivers of road bridges more easily. The structure works should be defined in more defined tasks according to Lee, Lee, Park, Choi, and Kwon (2016). Lee et al. (2016) divided the structure into a substructure and a superstructure.

3.6.5 Introduction to SSK

As mentioned in the introduction of paragraph 3.5, the available cost data that is used in this project is structured according the SSK method. These data are from cost estimates from former projects executed by Witteveen+Bos. SSK is only a format without real details of a project. This section further elaborates on SSK.

For the real detailed elements, the RAW standard, which is a large set of work descriptions, is applied in the SSK template. The installation of a wall of sheet piles according to a specific method can be such a RAW work description.

SSK uses a division of an asset into parts for clear communication and manageability. These parts are called objects. In principle, here an "object" is defined to be assembled into a whole from materials and parts. In the case of this project, the object is a road bridge.
Therefore, the object is real and has three dimensions. It is visible, touchable, and recognizable separately.
The object layout is of great importance for recording and drawing up the estimate. It is a way to communicate effectively with the stakeholders during the specification and design process about the estimate's content (CROW, 2018).

The selection of objects and the object classification of an asset is possible in one or more of the following ways:

- Geographical: the objects are distinguished or bundled because they are realized in certain places. On drawings, the location can be indicated through a line, an area, or a point.
- Process-based: the objects are bundled or distinguished because they are realized in different phases of the project.
- Work-based: the objects are bundled or distinguished based on the type of work. For example, the task ‘pouring concrete’ is clustered for all objects and phases of the project.
- Function-based: the objects are bundled or distinguished based on the function. For example, all different types of retaining walls.
- Contract type-based: the objects are bundled or distinguished based on the contract type. Selecting on contract type is of course usually only possible in the case of large projects for which several contracts are drawn up.
- Finance-based: the objects are bundled or distinguished based on price (CROW, 2018).

This bridge project is split into components. The available data consists of projects with components that are geographically segregated.

In the SSK, the cost overview and the underlying object estimates are divided into cost categories. The investment costs and the maintenance costs are both divided into the same cost categories. Each cost category consists of several cost groups. These cost categories are construction costs, engineering costs, property costs, additional costs, object transcending risk reservation, and uncertainty reserve (CROW, 2018). All cost groups together are the foreseen costs. These costs are the sum of appointed direct costs, direct costs to be detailed and indirect costs. Additionally, there is a risk reservation and VAT.

Direct costs
These costs are directly involved in the production or delivery of a product or service. These costs are directly attributable to the product or service and always consist of the three-unit: labor, (processing) plant, and material.

Allowances
These costs are a surcharge on the appointed direct costs for planned but not yet explicitly elaborated components. Depending on the phase of an asset, part of the design is not yet fully developed.

Indirect costs
The indirect cost group includes the costs within an object that are not attributed directly to a specific activity. For the indirect costs, it is impossible or too laborious to break it down per item. The indirect costs are therefore charged indirectly to the individual products or services. It is the sum of -one-off costs, execution costs, general costs for an asset, profit, risk, provisional sum, and premiums (CROW, 2018).

Boundaries
An SSK estimate has one price level, and the percentage price change is determined with index numbers. Market developments are not included in an SSK estimate. In principle, an SSK cost estimate therefore does not represent all the preconditions associated with a specific project.

Uncertainties
There are several uncertainties. Decision uncertainties can be solved to estimate each variant separately. Standard uncertainties arise from the lack of information.

Risks
Inventoried risks can be quantified. If the chance is >50%, they are calculated as an ordinary event. In the case of a probabilistic simulation, a spread of consequences is taken into account (CROW, 2018).

This paragraph made clear how the cost of a project can be divided using the conventional method. The database consists of projects which have already been executed by contractors. This means that not all cost categories are applicable.
In the reference bridge construction projects, no quantified risks are included besides fixed percentages in the overhead. The projects have no direct cost that should be further elaborated.

For a detailed description of the materials, equipment, deployment of personnel, and other costs, various standards can be used and implemented in the SSK format, including RAW.

3.7 Theory results model

These models are using datasets. In case the data from those datasets is used to calculate results, the results can be calculated by means of cross-fold validation. This is described in section 3.7.1. To check whether the model that calculates the results uses the most optimal setting, a hyperparameter analysis can be performed as described in section 3.7.2. Then there are also criteria against which the performance of models can be measured. This is described in section 3.7.3.

3.7.1 Cross-validation

Cross-validation is a method for datasets to calculate the test results. Leave-one-out cross-validation, is a particular case of cross-validation, and often used for small datasets, in which each instance is used once as the test case, and all other instances are used as the training set (Syed, 2014). Then are the results for the test sample calculated by the model. Then, the next sample is the test sample and the other samples are the training samples and results are calculated. This process is repeated until each sample has been tested.

When a dataset contains n projects and n-fold cross-validation is used, n-1 projects are in the model's database, and the nth project is the test case. This method is also called n-fold cross-validation. In the case of 10-fold cross-validation and a dataset with a number of n projects, the dataset is split into 10 samples of n/10 projects. Figure 9 shows an example of 10-fold cross-validation. The average value E of the ten-groups test results is calculated as an estimate of the model accuracy and is used as a performance indicator for the current K-fold cross-validation model. Where Ei represents the cross-validation error of the ith group (Niu, Li, Wang, & Han, 2018).

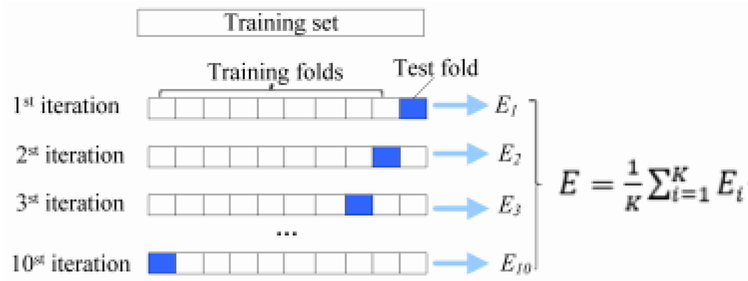

Figure 9: 10-fold cross-validation (Niu et al., 2018)

5-Fold cross-validation and 10-fold cross-validation are often used to determine a prediction error. Then, the larger sample size is used because of the computational time which can add up enormously in the case of large datasets (Fushiki, 2011).

3.7.2 Hyperparameter analysis

A machine learning model is normally able to use different parameters in the model itself to solve a problem. For example, in case of FALCON, the model can use different membership functions. These parameters are called hyperparameters and these parameters affect how the machine learning model results fit to the desired result. In order to obtain the best results with a model, the best set of hyperparameters should be known. The process of finding the best set is called hyperparameter tuning or an hyperparameter analysis. The hyperparameter analysis can be done manually or by using an automatic search algorithm as Grid search (Wu et al., 2019).

3.7.3 Performance of machine learning models

Evaluation of the performance of machine learning models is possible using the performance measures as described by Østergård, Jensen, and Maagaard (2018).
These performance measures are:

- Accuracy
- Interpretability
- Robustness
- Ease of use

The accuracy describes the deviation from the real values that should be approximated by a model. The interpretability refers to the amount of insight that a model gives into the model behaviour. Robustness refers to the ability to produce acceptable results independent on diverse problems that can occur. The ease of use relates to the process of implementing the model.

4 Cost estimation model development

In chapter 3, detailed information about the FALCON model, KNN model, decision tree model, and multiple linear regression model is presented. In addition, detailed information about data structuring, and theory regarding the results is presented.

This chapter describes the step from theory to an artefact, a cost estimation model for road bridges. In this chapter, 4 models are presented that all use the same data. The same data is used because of the comparability of the results calculated by the models.

This chapter aims to present the best solution for the 'construct requirement': the best performing model that is suitable to make quickly the most accurate cost estimates for bridge projects. This requirement is already presented in the research strategy in chapter 2.

Paragraph 4.1, and paragraph 4.2 focus on the data that the models use. Thereafter, paragraph 4.3 is about the FALCON model. In paragraph 4.4, the multiple linear regression model is elaborated, followed by the decision tree regression in paragraph 4.5. The fourth model, K nearest neighbors is elaborated in paragraph 4.6. The final paragraph 4.7 presents the verification of the solutions to the requirement, as mentioned above. In this paragraph, it is concluded which model gives the best solution.

4.1 Input cost

Section 3.6 described several aspects related to the structuring process of the input data: the use of an index, determining a level of detail and the work breakdown structure. This section brings these aspects together in order that the cost data have been structured by category and then a database is created that can be used by the FALCON model.

The database contains projects with different reference years. The CBS index has been used to equalize the price level to the price level of 2020. This index is described in section 3.6.1. Appendix I shows all index numbers of the CBS index.

The reference projects contain a lot of detailed cost information. In the early phase of a completely new project, not enough information is available to come up with a prediction with a high level of detail. In this research project, the level of detail is somewhere between level 2 and level 3, as those are presented in section 3.6.3. Initially, Level 2, the heading level, is assumed since this level fits all projects.

Section 3.6.4 showed some examples on heading level. A work break down structure for concrete bridges and a work break down structure for steel bridges is shown in that section. First these divisions are combined to a list for steel bridges and concrete bridges. For this research project, the list should also be suitable for other bridge types and therefore ends up in:

1. Preparatory work
2. Drainage work
3. Soil work
4. Pavement widening and shoulder widening works.
5. Pavement and hardening coating.
6. Structure works
7. Daily work

Lee et al. (2016) divided the structure works as mentioned in section 3.6.4 into a substructure and a superstructure. Within the substructure, sub-levels with more detailed information can be defined: the substructure and foundation structure. The foundation structure contains the structural parts of the bridge underground such as the foundation piles. The substructure contains the structural parts between the foundation and the bridge deck as seen from bottom to top. Among others, the abutments and pillars are part of the substructure. Also, within the superstructure, sub-levels with more detailed information can be defined: the superstructure, the railing and external finishes. The superstructure consists mainly of the bridge deck without additions. The railing contains exactly what the name suggests, only the railing. The external finishes are mainly matters related to the layout of the site in completed condition, as also mentioned by Wang et al. (2017). In addition, there is a category for several aspects that cannot be specifically assigned to the aforementioned categories. This category with general work is called general cost. The resulting break down structure is listed below.

1. Preparatory work
2. Soil work
3. Foundation structure
4. Substructure
5. Superstructure
6. Pavements
7. Railings
8. External finishes
9. General cost

The nine mentioned cost divisions are defined for all projects. All division levels have different factors that are affecting them.

These cost division categories are related to the direct building costs, not to the indirect costs as profit and insurances. These costs are part of the overhead in the method which Witteveen+Bos uses for all his cost estimates. This method is called SSK and is already described in section 3.6.5. Therefore, the category overhead (in Dutch: staartkosten) is added to the list above.

10. Overhead

All these categories will be further elaborated and explained. Further elaborated categories create the possibility to calculate the costs per category.

As mentioned in the SSK introduction, section 3.6.5, the RAW standard is used for accurate work description. RAW is one standard in which describes all activities for bridge construction projects. Besides, this standard is widely accepted in the Netherlands. These descriptions are universal, and contractors know their meaning. Not all descriptions of activities in the RAW standard are suitable for each project. For each project, a selection is made of the activities.

The activities as presented in the RAW standard, version 2020, and applicable to the categories for bridge projects are listed per category and presented in Appendix II. The RAW standard is owned by the Dutch Organization CROW and is written in Dutch. The list in Appendix II is written in Dutch.

A couple of categories has extra descriptions with parts that are relevant to that category. For the foundation the drainage, construction pit, and sheet piles have been included in that part, (in Dutch: kuip, and damwanden). In the substructure are the pillars, floor, land abutment, wing wall, girder, supports (blocks), capping beam, foundation blocks, and retaining wall have been included, (in Dutch: pijlers, vloer, landhoofd, vleugelwand, ligger, opleggingen, deksloof, poeren, and keerwand). In the superstructure the bridge deck edge, floor, anchors of the railing, joint construction, curb have been included, (in Dutch: randen, vloer, ankers (leuning), voegconstructie, and schampkanten).

In Appendix II, all categories, including the RAW categories, are listed.

This classification is used to determine the costs per category per bridge. This classification is, however, not the same as the input data of the FALCON model. After classifying the projects from the database according to the categories, the following division of percentages based on the total price of projects appears. Figure 10 shows the average percentages of all projects per category based on costs.

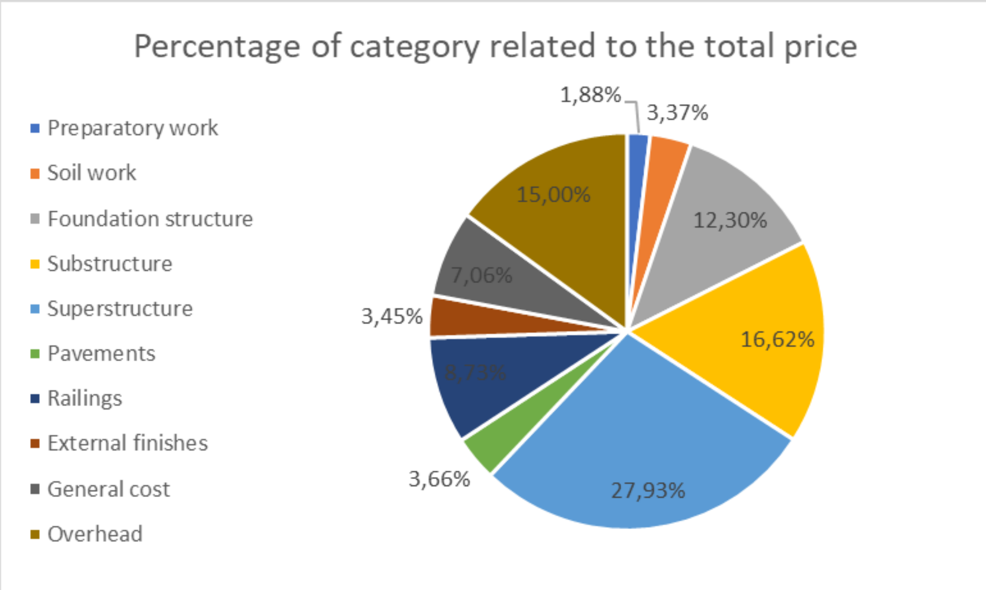

Figure 10: Percentage of category related to the total price

Based on this division, the four most important categories according to the presented percentages, the foundation structure, substructure, superstructure, and railing are selected as input for FALCON. The model makes an estimate for these four categories. The estimate must be scaled to 100%. This is equal to the approach of Wang et al. (2017), who used also FALCON for cost estimates of constructions. The next paragraph, 4.2, focuses on the input variables for FALCON. The input variables are used by the FALCON model to come up with a cost estimate.
In the sections before, the classification of the reference projects' data is described. The four models must calculate a price based on properties of a project. More specifically, the properties of the four main categories (foundation structure, substructure, superstructure, and railing).

As marked in the sections before, there are four major categories concerning the cost of a bridge. These categories (foundation structure, substructure, superstructure, and railing) are directly related to the building process.

Another major category according to the percentages in figure 10 is overhead. These costs are more general and not specifically related to one building item and are a percentage added to the other costs.

The overhead is implemented in the input. For each project, the overhead is divided pro-rata over the other categories. A larger percentage of the total costs of each project can thus be included in the four models. Figure 11 shows the new division. For each reference project, the costs of the 4 main categories according to the new classification are used as input for the reference database that the models use. All data is adjusted to the same index year, which is 2020.

This only concerns the price of euros per m2 per category. Section 4.2 discusses the properties per project that will be added to the cost data.

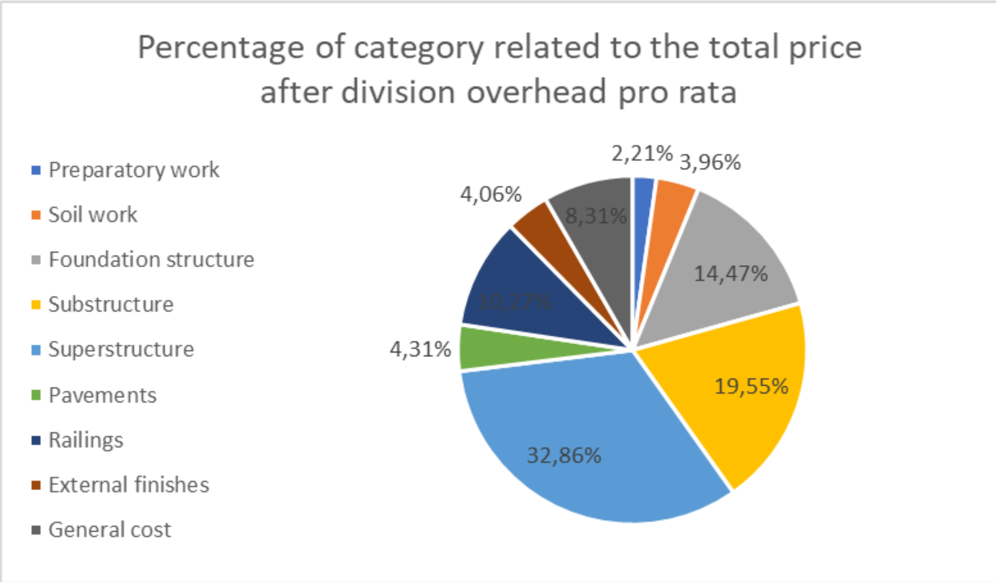

Figure 11: Percentage of category related to the total price after division overhead cost pro-rata

4.2 Input properties

As mentioned in chapter 1, less information is available in the early phase of a project. The models make a prediction based on this information. The information that serves as input to the four models and is used to compare projects must therefore be structured. The simplest way is a short 'Google' form with 'questions' about the 'new' bridge's properties. The short form makes the model easy to use, and it takes little time. The form concerns the presence of parts or having a particular characteristic.

Of course, there is a set of properties that generally belong to a bridge. Some of the parameters are only related to one of the four main categories. The general parameters are first set and apply to the total set parameters of a subset.

The general parameters do not necessarily have much information about the costs but indicate the basis of the bridge's 'DNA.'
The parameters of the four detailed parts of the bridge result from a cost driver analysis of the project data and the division of bridges on project type as mentioned in section 3.5.2. Based on this data, it became clear, what the cost drivers per category are. Based on these cost drivers, the parameters for these parts are selected.

These parameters, as shown in table 3, are defined for each reference bridge and added to the dataset (true/false/number). Some of these parameters are general and some of them are related to a category.

Table 3: inputparameters FALCON model

General parameters	Foundation structure	Substructure	Superstructure	Railing
Movable bridge	Steel piles	Wing walls	Bridge deck steel	Wood
Concrete road bridge	Concrete piles	Land abutment	Bridge deck prefab concrete	Steel
Concrete bicycle/pedestrian bridge	Steel piles filled with concrete	Temporary floor	Preservation steel	Wood and steel
Wooden road bridge	Remove pile heads	Concrete formwork	Compile bridge deck elements	Artwork
Wooden bicycle/pedestrian bridge	Drainage	Baffle plates	Use temporary construction	Brickwork
Steel road bridge	Sheet piles	Bearing blocks	Brickwork superstructure	
Steel bicycle/pedestrian bridge	Use construction pit	Reinforcement abutment	Engineering structure	
Plastic bridge	Armed ground	Reinforcement retaining walls		
Other bridge type		Concrete in abutment		
Material wood		Concrete in retaining wall		
Material concrete		Completely prefab		
Material steel				
Material steel – concrete				
Bridge deck area in square meters				
Amount of intermediate supports				
Length railing in meters				

4.3 FALCON

This paragraph elaborates the FALCON model. First in section 4.3.1, the step form theory to a model is made. Next in section 4.3.2, the results generated by the FALCON model are shown. Thereafter, in section 4.3.3, it is shown if the results are improved after a hyper parameter analysis. In the last section, section 4.3.4 the total list of findings is presented.

4.3.1 Theory implementation

Paragraph 4.1 and 4.2 described the construction of the dataset which is used for the FALCON model. In this section, the construction of the model, according to the literature, is described. The explanation is done per layer, starting at the bottom of the network. The scheme below shows an overview of the FALCON model. The FALCON network is programmed in Python; Python is an open-source programming language. This chapter describes the FALCON model. Chapter five discusses the results obtained with this model.

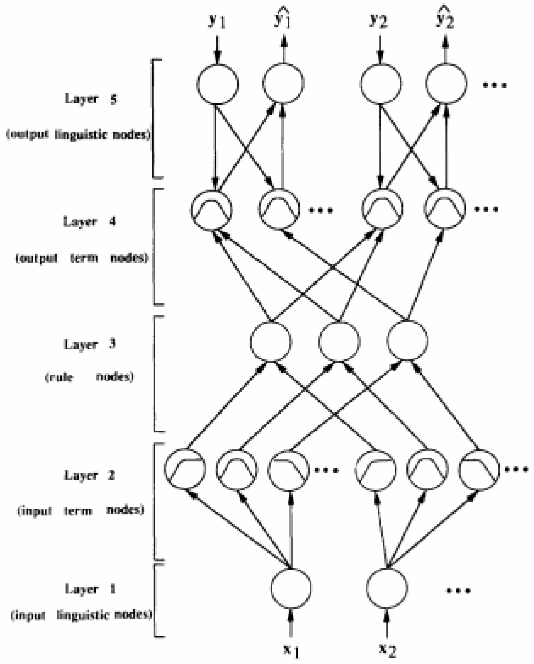

Figure 12 Schematized representation of FALCON (C.-J. Lin & Lin, 1997)

4.3.1.1 Layer 1

In the first layer, as seen from bottom to top in figure 12, the fuzzification of the input takes place. Layer 1 just transmits the input values directly to the next layer (C.-J. Lin & Lin, 1997). In practice, the input sheet fills in this stage by ticking the true boxes and adding a couple of numbers.

The output of layer 1 is a couple of arrays with values. These values are the same as entered on the form above. For each specific part, category, the script forms a new array. These arrays are the output of layer 1 and the input of layer 2.

4.3.1.2 Layer 2

In layer 2 as shown in figure 12, the fuzzification of the data takes place. Fuzzification means that categorical data transforms into numerical data and that numerical input is scaled. In the end, all input transforms to values from zero to one.

As mentioned above, not all data has the same format, which means that more than one membership function transforms all data into numerical data.

For numerical data, the FALCON model uses a triangular membership function. The parameters of a triangular function are a, b, c, and the variable x. In this function, a, b and c will represent values from small to large.

Equation 3: Triangular function

$$f(x) = \begin{cases} 0, & x \leq a \\ \dfrac{(x-a)}{(b-a)}, & a < x < b \\ \dfrac{(c-x)}{(c-b)}, & b < x < c \\ 0, & x \geq c \end{cases}$$

For categorical data, the FALCON model uses a Heaviside membership function.

Equation 4: Heaviside function

$$Heaviside\ (x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

The output for each category is an array with numbers between zero and one.

4.3.1.3 Layer 3

The connections between layer 2 and 3 represent the fuzzy If-then rule preconditions. The 'fuzzy AND' operation follows thereafter. The last step is the link between layer 3 and layer 4. This link represents the 'THEN' part.

In practice, this means that if rules are generated, as expected, layer 3 uses the output from layer 2 as input for the layer. For the IF-THEN rule precondition, this is illustrated with an example. IF-statements were created; if parameter 100 = 0, 101 = 0 and 102 = 0 this gives an array [100, 0, 101, 0, 102, 0]. These IF-statements vary in length.

The AND operation finds the best match between all projects in the database and the IF-statement.
'THEN,' the final step for this layer is the final link between the input, the IF-statement, and the best matching project. Per length of elements, the python script generates the best match.

Such a score consists of a couple of items. First, the number of elements, secondly the agreement scores of the input parameters with the model parameters. Thirdly the best combination and the best matching project. For example: for 7 elements, a 100% match is with reference project 3. The 100% match is for combination set 83.

This output is not the final output and is the pre-result for one of the four main categories. The next layers take a few more steps to arrive at the final output.

4.3.1.4 Layer 4

In this layer, the OR operation takes place. Based on the output of layer 3, the model selects the best. The OR operation is done based on the final score. This final score is calculated by multiplying the amount of elements times the score.

For example, 9 * 90 = 810, 10 * 88 = 880, 11 * 80 = 880. The total sum of the last two is equal. Suppose the total sum does not increase when an extra (scoring) element is added. Therefore, that element does not have a (partly) corresponding value or the average agreement per element has decreased. In such a case, the model selects the project that belongs to the total score of 880 with 10 elements.

Now, the best matching project is selected. In the case of the main categories; foundation, substructure, or superstructure, the model multiplies the price in the euro per m2 of the foundation, substructure, or superstructure of the reference project with the number of square meters of the new case gives the final partial result. In the case of the main category railing, the multiplication is done per meter length of the railing

The output consists now of 4 partial prices, and they are the input of layer 5.

4.3.1.5 Layer 5

This layer produces the final results. The result is a summation of the four category results, (foundation, substructure, superstructure, railing). This amount is divided by 77 and multiplied with 100 and produces a total estimate for the project. This short calculation includes the categories that are not in the Python model, although they cause a part of a project's total price.

4.3.2 Results

As mentioned in the former chapters, the FALCON model generates a price for each of the four main categories and a total price. This total price is most important because this price is the total estimate of the tender. Witteveen+Bos is looked upon as the tender price is not within the set bandwidth. The prediction of Witteveen+Bos has a strong influence on the budgeting of a project by the client of Witteveen+Bos. In the case of a misfit, the client is willing to hire another company for a new project.

A method for small datasets is used to calculate the test results. This method, Leave-one-out cross validation is described in section 3.7.1. A 39 cross-fold validation has taken place. Figure 13 visualizes the absolute errors. The numerical values of figure 13 are shown in table 4. The absolute errors are the absolute errors between the prediction of the FALCON model and the real value of that project in the database. These errors are visualized for each of the categories and the total error. The shown data is corrected for outliers. For the visualization of the data are, 33 projects were used. Appendix III shows all output per project.

The influence of the size of the dataset on the predictions of a model is stronger when a model uses a small dataset than when a model uses a large dataset. The FALCON model must generate results in the most optimal conditions. In the case of n-fold cross-validation, the test sample can be compared with the largest number of projects. Therefore, the best results are expected for this type of cross-validation. To check if this assumption is correct, 5-fold and 10-fold cross-validation are also performed. 5-Fold cross-validation and 10-fold cross-validation are often used to determine a prediction error. Then, the larger sample size is used because of the computational time which can add up enormously in the case of large datasets (Fushiki, 2011).

Using 3 types of cross-validation, the dependence on the dataset size is shown in case FALCON is used to make cost estimates for bridges. This made clear if this variable, the dataset size has some influence on the robustness of the FALCON model.

The results of these cross-validations are comparable to n-fold cross-validation. As expected, the performance of the FALCON model in the case of n-fold cross-validation is slightly better. All results, including the results of 5-fold and 10-fold cross-validation, are presented in Appendix III.

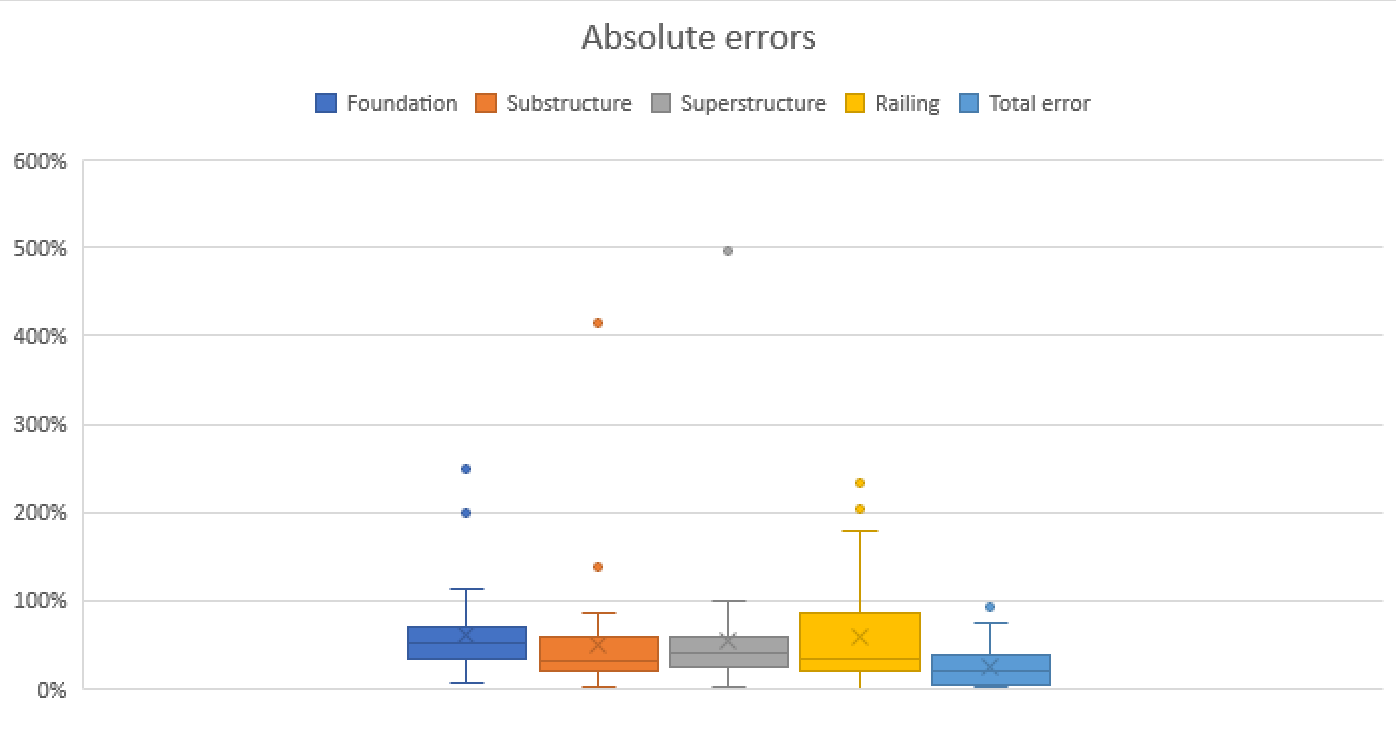

Figure 13: Absolute errors FALCON n-fold cross-validation

Table 4: Values Boxplots FALCON n-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Values above maximum	250%	416%		234%	
Values above maximum	199%	138%	497%	203%	93%
Maximum error	113%	87%	99%	179%	75%
Upper Quartile	71%	58%	58%	85%	39%
Average error 'X'	61%	49%	54%	58%	24%
Median	52%	30%	40%	34%	20%
Lower Quartile	34%	21%	24%	20%	4%
Minimum error	7%	2%	1%	0%	1%

4.3.3 Hyperparameter analysis

In section 3.7.2, the hyperparameter analysis is explained. This concept is used to find the most optimal setting of the FALCON model to find the results. It checks if the parameters used in the former section are responsible for the best results with this model. In case of FALCON, only one parameter must be tuned, therefore it is possible to do the hyperparameter analysis process manually.

As mentioned, in the case of FALCON, the membership functions can change into other membership functions. Commonly used membership functions are the impulsive membership function (Heaviside), triangular membership function, right-sided trapezoidal function, left-sided trapezoidal function, and the Gaussian membership function (Ling, 2010).

The FALCON model uses two of the above mentioned two functions. The triangular function and the Heaviside function. Except for the impulse function, all functions are continuous functions. The Heaviside function is discontinuous, and the data is also discontinuous. Therefore, placing a continuous function through discontinuous data, true and false, will not be very promising for improving the results.

In the FALCON model, the triangular function is used and modelled only for one side of the triangle. This function is equal to the left side trapezoidal function, and the reflection will be the right-sided trapezoidal function. These functions are therefore not very promising to replace the triangular function.

This function is applied instead of the triangular membership function, and results are generated. As shown in figure 14 and table 5, the results with a Gaussian membership function will not improve the results, and the average error will increase to 32% instead of 24% in case of n-fold cross-validation. A complete table with all results per project is available in Appendix III.

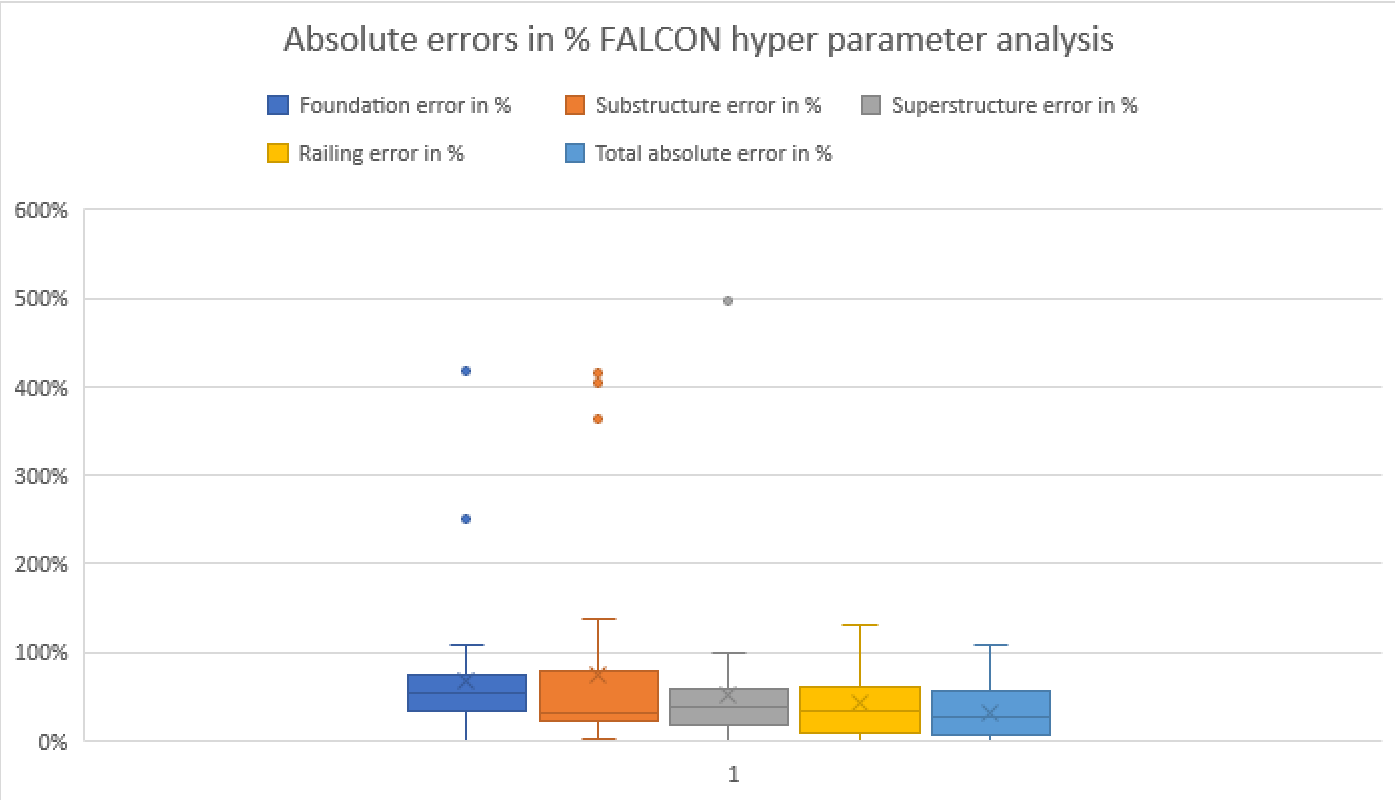

Figure 14: Absolute errors FALCON hyperparameter analysis

Table 5: Values Boxplots FALCON hyperparameter analysis

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum	418%	403%			
Value above maximum	250%	362%	497%		
Maximum error	107%	138%	99%	130%	109%
Upper Quartile	74%	79%	58%	62%	57%
Average error 'X'	68%	75%	53%	43%	32%
Median	54%	31%	38%	34%	26%
Lower Quartile	33%	22%	18%	9%	6%
Minimum error	0%	2%	1%	0%	0%

4.3.4 Findings

Based on the results presented in sections 4.3.2 and 4.3.3, this section elaborates on the findings using performance measures as described in section 3.7.3. These performance measures are the accuracy of the predicted total price, described in section 4.3.4.1, the accuracy of the predicted price for components, described in section 4.3.4.2, the interpretability of the results, described in section 4.3.4.3, the robustness of the FALCON model, described in section 4.3.4.4, and the ease of use, described in section 4.3.4.5. Besides some boundary conditions have been described in section 4.3.4.6.

4.3.4.1 Accuracy of the total price

The figures and tables in sections 4.3.2 and 4.3.3 showed the results, including the partial results of each of the four categories. As shown in table 4, the absolute average error on the total price is 24% in the case of n-fold cross-validation.

The outcome is based on comparable projects, as also stated in section 3.5. The fact that the average deviation is 24% indicates that the projects are reasonably comparable. If the projects were not comparable at all, the deviation would vary more.

However, if FALCON is combined with fmGA, the error is 16.18% in piping projects, which is mentioned in paragraph 1.5 by Hsiao et al. (2012). In paragraph 1.5, it is also mentioned that (Wang et al., 2017) used FALCON for building projects. An average error of 9.72% is reached for building projects based on three projects (19.98%, 5.84%, 3.33%), the error in road bridge projects is larger.

4.3.4.2 Accuracy components

In addition to the accuracy of the model on total price, it is also essential to have insight into the parts' figures as presented. The total price errors are not as large as the error average error of one of the four categories (foundation, substructure, superstructure, railing) would be. There are a couple of reasons for these differences:

- First, positive and negative deviations within a project can partially offset each other.
- Besides, the percentage of one of the four components in the total price is not equal. For example, a large deviation in the railing does not count as much as a large deviation in the superstructure in most cases.
- Other options and reasons for large errors are the reference projects. In case this project is not that expensive compared to all similar projects in the database, (based on the input of the FALCON model), a large deviation on price would probably the result. At this stage, considering the number of projects in the database, a division between cheap and expensive projects based on a threshold cannot yet provide enough added value.

4.3.4.3 Interpret the results

Another topic is the interpretability of the results. Without context, it is not easy to put the numbers in perspective. According to the results, FALCON should be a powerful tool for the prediction of cost for bridge projects.

Besides, the model can provide broad insights into the cost drivers of a project. The operation of the model itself can be clearly explained and is explained as such.

In addition to all information in the FALCON model, many factors are responsible for the prediction of price. Considering factors such as economic factors, environmental factors like protesting neighbours, NGO's, political decisions, and the presence of special animal species. That many factors influence the price means that the FALCON model does reasonably not include all these characteristics to arrive at the prediction. Besides, cost experts can understand the FALCON model, and therefore adjustments are possible to improve the model continuously.

4.3.4.4 Robustness

Robustness must also be considered. Concerning the FALCON model, robustness is related to errors and how to deal with them. Errors can be in the model's input, although they can occur in the execution of the model. The results show that the influence of the sample size in the used small dataset is not that much.
Concerning the input data, the datasheets before the input were defined and are mostly generated by a computer program; the number of errors should therefore be negligible. The next step, from sheet to a category, is done by hand, the chance of errors increases. Although extensive controls after this phase have kept chis chance to a minimum. Related to the model's input, it is impossible due to the drop-down menus to add wrong data in the wrong field.

4.3.4.5 Ease of use

Ease of use is another performance measure. The current FALCON model can of course be used by anyone without knowledge of the data or programming. This is a big advantage. In the end, the FALCON model results should be evaluated by an analyst with some knowledge of cost engineering.
In case the model will be used for other project types, the development process will go smoothly in case the developer has knowledge of the FALCON model.
Selecting projects and generating input data for the database is the first step that employees can do without knowledge of Python or models like FALCON, therefore a simple manual should be sufficient.

4.3.4.6 Boundary conditions

This model is based on the cost drivers of 39 reference projects, which means that the model considers the items that are responsible for a significant amount of the cost in these projects. Economic factors, stakeholder requirements, and resource factors are no parameters in the model. When one of these parameters is responsible for a significant part of a bridge project's price, the prediction accuracy is also influenced negatively by this significant part. In practice, the cost estimator considers such parameters. However, it is almost impossible to take such project-dependent relations into account. These project-dependent relations can be mapped out using risk analysis, followed by measures. A cost estimator can determine the price of these measures. Besides, the database consists mainly of that are fixed concrete or steel road bridges. These bridges are part of the secondary road structure. The model will, therefore, when the database of Witteveen+Bos is used, only apply to these types of projects.

4.4 Multiple linear regression

This paragraph describes the multiple linear regression. This paragraph starts with theory implementation in section 4.4.1, thereafter the hyperparameter analysis in section 4.4.2. The numeric results of this method are presented in section 4.4.3. Section 4.4.4 presents the findings of this method.

4.4.1 Theory implementation

Excel has a multiple linear regression module based on the least squares method. Solving the problem with Excel is very accessible to everyone. However, this module has a limited capacity of 12 parameters. This number of parameters is less than the total amount of parameters in the database. Therefore, a calculation is made in Python. Python has a multiple linear regression module. This module has a couple of parameters that can be tuned by hyperparameter tuning.

4.4.2 Hyperparameter analysis

Before generating the results, first, a hyperparameter analysis is executed. Besides the data, the Python module has some parameters which can be tuned. Using the grid search module in Python is used to make an analysis and select the best parameter.

Section 3.7.2 explained how hyperparameter tuning works. In that case, results were generated for the model to determine which parameter works best. The models that are tested in this chapter are all programmed using a Python module. Therefore, these models are suitable for use of the Grid Search module. As mentioned in section 3.7.2, Grid Search is a module for quickly determining the optimal set of parameters so that a model calculates the best possible results. This ensures that the final results, the predicted total price of a bridge project, only need to be calculated once and not for every possible combination of hyperparameters.

In total, two tuning parameters are available: fit intercept and normalize.

For the second parameter, it is obvious what the possibilities are. This parameter specifies whether to normalize the data before executing the multiple linear regression. The factor fit intercept is related to the factor ‘E’ as shown in the multiple linear regression formula. If the factor fit intercept is true, a factor 'E' is part of the formula to calculate the results. In case the factor is set to false, no factor 'E' is part of the multiple linear regression formula.

According to n-fold cross-validation, the optimum is defined for each parameter in each fold. To determine the optimum setting, the test sample is not part of the data. The test sample may not have been used to set the model in the most optimal setting. This optimum is defined for each category. Table 6 shows the optimum parameters in the case of n-fold cross-validation.

Table 6: Optimum parameters multiple linear regression analysis

	Foundation		Substructure		Superstructure		Railing	
	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize
Result	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE

4.4.3 Results

The next step is the calculation of results for n-fold cross-validation. As mentioned in section 4.3.2, in the case of n-fold cross-validation, the test sample can be compared with the largest number of projects. Therefore, the best results are expected for this type of cross-validation.

These results are corrected for outliers of the total price.
After omitting the outliers, the results in figure 15 and corresponding table 7 visualize a total of 33 projects. Some values above or below a boxplot maximum or minimum still occur. Table 7 shows these values. An n-fold cross-validation setting gives the best results. In case the results are generated with 5- or 10-fold cross-validation, the results are not that good. These results are documented in Appendix IV. The results per project, including reasons for the omission of outliers, for the n-fold cross-validation are also documented in Appendix IV.

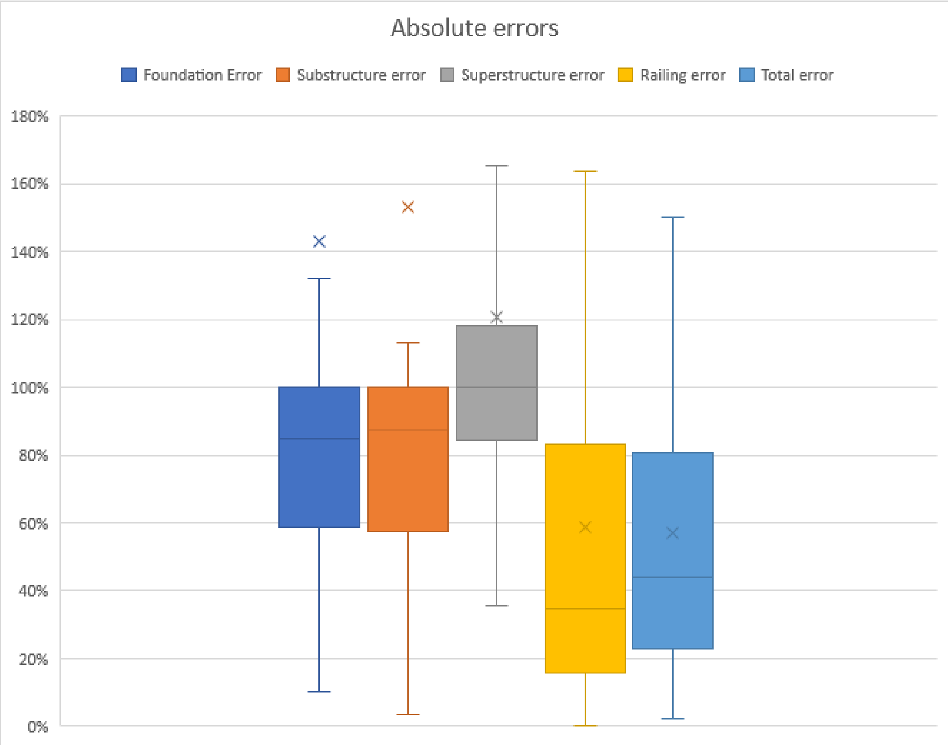

Figure 15: Absolute errors multiple linear regression n-fold cross-validation

Table 7: Values boxplots multiple linear regression n-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum		1220%			
Value above maximum		683%	495%		
Value above maximum		629%	470%		
Value above maximum	1970%	267%	249%		
Value above maximum	218%	254%	224%	359%	176%
Maximum error	132%	113%	166%	164%	150%
Upper Quartile	100%	100%	118%	83%	81%
Average error 'X'	143%	153%	121%	59%	57%
Median	85%	87%	100%	35%	44%
Lower Quartile	59%	58%	84%	16%	23%
Minimum error	10%	4%	35%	0%	2%
Value below minimum			26%		
Value below minimum			9%		

4.4.4 Findings

Based on the results presented in sections 4.4.2 and 4.4.3, this section elaborates on the findings using performance measures as described in section 3.7.3. These performance measures are the accuracy of the predicted total price, described in section 4.4.4.1, the accuracy of the predicted price for components, described in section 4.4.4.2, the interpretability of the results, described in section 4.4.4.3, the robustness of the FALCON model, described in section 4.4.4.4, and the ease of use, described in section 4.4.4.5. Besides some boundary conditions have been described in section 4.4.4.6.

4.4.4.1 Accuracy of the total price

The figures and tables in sections 4.3.2 and 4.3.3 showed the results, including the partial results of each of the four categories. As shown in table 7, the absolute average error on the total price is 57% in the case of n-fold cross-validation. The average error is larger than the errors in case an estimate is made by a conventional method. The values are in that case between 15-50% as presented in chapter 1.

4.4.4.2 Accuracy components

In addition to the accuracy of the model on total price, it is also essential to have insight into the parts' figures as presented. The total price errors are not as large as the error average error of one of the four categories (foundation, substructure, superstructure, railing) would be. Given the enormous deviations, it can be assumed that the linear relationship in these data is not present enough to be able to plot a linear line through the data points of the dataset.

4.4.4.3 Interpret the results

In the case of multiple linear regression, the interpretation of the results is relatively easy because it is possible to get the formula that was used to calculate the result. However, given the deviations from the model, it is difficult to give value to the results.

4.4.4.4 Robustness

Robustness must also be considered. Concerning the multiple linear regression model, robustness is related to errors and how to deal with them. Errors can be in the model's input, although they can occur in the execution of the model. The results in Appendix IV show that the influence of the sample size in the used small dataset is large. The results in Appendix IV show the average errors of the other cross-validation types. The average error for 10-fold cross validation is 67%, in case of 5-fold cross validation is the average error 68%. From this it is concluded that the size of the dataset has a significant influence on the performance of this model.

Concerning the input data, the datasheets before the input were defined and are mostly generated by a computer program; the number of errors should therefore be negligible. The next step, from sheet to a category, is done by hand, the chance of errors increases. Although extensive controls after this phase have kept chis chance to a minimum.

4.4.4.5 Ease of use

The ease of use of this model can be assumed to be relatively good. There is a module available to program the model. Using the Python module results in a programming process in which not every step has to be programmed by the programmer of the model. However, the model takes the projects from the datasheet, which means that there is no input sheet like a Google form that makes it very easy to use the script in a new project. This should be added. The runtime of the script is fast. Within a minute, the results are produced.

4.4.4.6 Boundary conditions

As known from the FALCON model, economic factors, stakeholder requirements, and resource factors are also no parameters in this model. That means that this model is not 100% focused on a specific case, including all the associated factors.

4.5 Decision tree regression

This paragraph describes the decision tree regression. This paragraph starts with theory implementation in section 4.5.1, thereafter the hyperparameter analysis in section 4.5.2. The numeric results of this method are presented in section 4.5.3. Section 4.5.4 presents the findings of this method.

4.5.1 Theory implementation

The output of this model should be a continuous variable, a predicted price, and no class, for example, a bridge type. Therefore, a decision tree regressor is programmed in Python, instead of a decision tree classifier. This regressor generates a cost estimate for the total project price.
As input, this model uses the same database as used for FALCON, multiple linear regression, and KNN. In the script, the data is split into a test and a training set.
In this case, results are generated with n-fold, 5-fold, and 10-fold cross-validation used, equal to the result generation with FALCON. The estimated price compared with the actual price gives the error of the test sample.

Besides, the Python script generates a figure of the decision tree. An important note is that the decision tree can vary per sample since the splits are based on the training data.

Figure 16 shows one of the decision trees. The splits are based on impurity as earlier mentioned, the splits do not divide the data into equal parts. The mean square error is represented by mse. 'Samples' is the number of projects, in the case of this example, 7 of the 39 projects are omitted by the first node.

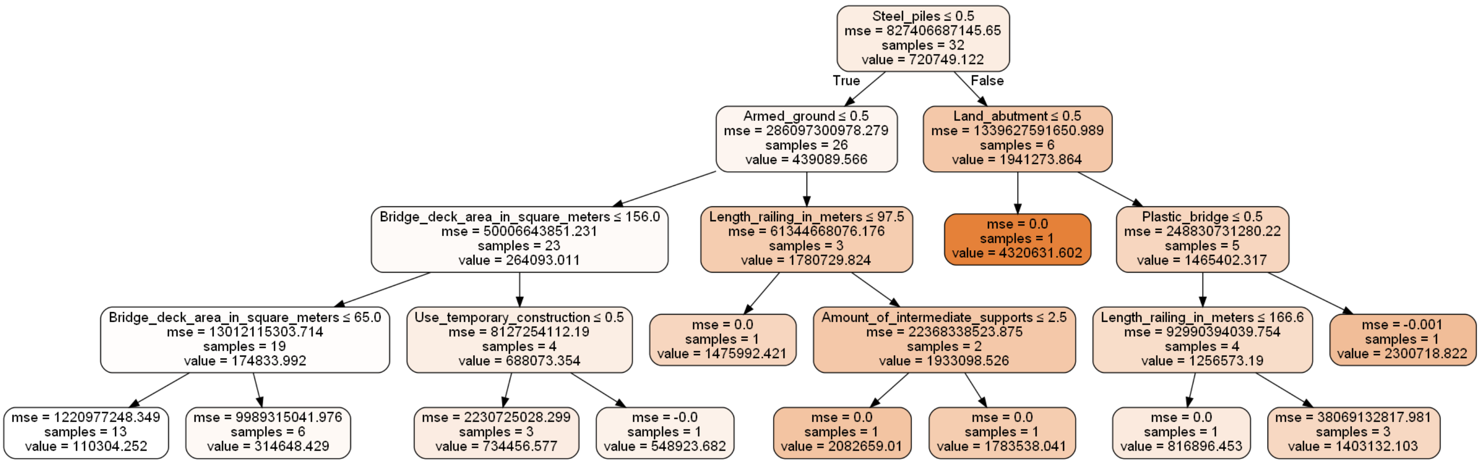

Figure 16: Decision tree

4.5.2 Hyperparameter analysis

Before generating the results, first, a hyperparameter analysis is executed. The model is programmed in Python. This Python script has some tuning parameters besides the data. Using the grid search module in Python, the best parameters are selected.

Three parameters are tuned: the depth, the minimum number of samples in one leaf, and the minimum number of samples for a split. A leaf is a node at which the tree stops splitting.

The depth means how many layers are below the root of the tree. The minimum samples in a leaf mean the number of samples (projects) in a leaf where the tree will stop. The minimum number of samples for a split means how many samples are necessary to have a split.

The optimum is defined for each parameter in each fold of the n-fold, 5-fold, and 10-fold cross-validation. To determine the optimum setting, the test sample is not part of the data. The test sample may not have been used to set the model in the most optimal setting. On average, it came out that the optimum parameters are the max depth = 4, the minimum samples for a split = 2, and the minimum samples per leaf = 1. The table with the results of the hyperparameter analysis is available in Appendix V.

4.5.3 Results

The next step is the calculation of results for n-fold cross-validation, these results are visualized in figure 17. As mentioned in section 4.3.2, in the case of n-fold cross-validation, the test sample can be compared with the largest number of projects. Therefore, the best results are expected for this type of cross-validation.

These results in figure 17 are corrected for outliers. The boxplot below visualizes thirty-five projects.

As can be seen in figure 17, an estimate has only been made for the total price of the bridge. Results have been also generated for the four separate categories, after which the total price was determined based on the results for these categories. However, the results of these calculations were worse than if the total price is determined in one go, which is after all the goal. Therefore, the results of every separate part are omitted.

Four projects are marked as an outlier. Project 9 is marked as an outlier because it is a unique project since it is a wooden bridge. Project 15, 25, and 29 are marked as extreme outliers and therefore removed.

The lowest value is 6%, the lower quartile value is 15%, the median 23%, the average value is 36% (x), the upper quartile value is 57%, and the highest value is 103%.

All results for each type of cross-validation are documented in Appendix V. The results per project, including reasons for the omission of outliers, for the n-fold cross-validation are also documented in Appendix V.

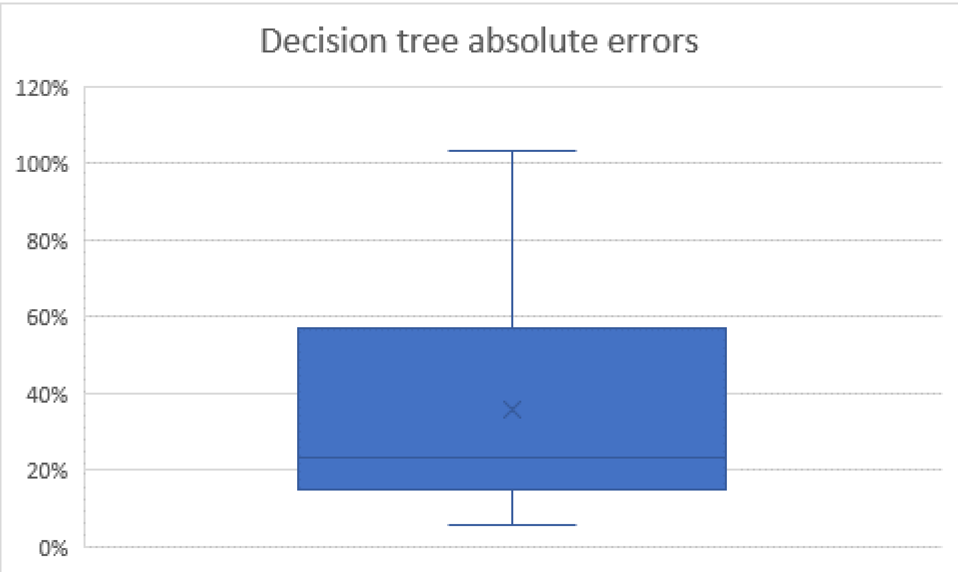

Figure 17: Absolute errors decision tree, n-fold cross-validation

4.5.4 Findings

Based on the results presented in sections 4.5.2 and 4.5.3, elaborates this section on the findings using performance measures as described in section 3.7.3. These performance measures are the accuracy of the predicted total price, described in section 4.5.4.1, the interpretability of the results, described in section 4.5.4.2, the robustness of the FALCON model, described in section 4.5.4.3, and the ease of use, described in section 4.5.4.4. Besides some boundary conditions have been described in section 4.5.4.5.

4.5.4.1 Accuracy

The figures and tables in sections 4.5.2 and 4.5.3 showed the results, including the partial results of each of the four categories. As shown in figure 17, the absolute average error on the total price is 36% in the case of n-fold cross-validation. The average error is in the same range as the errors in case the estimate is made by a conventional method. The values are in that case between 15-50% as presented in chapter 1.

4.5.4.2 Interpret the results

In the case of the decision tree, the interpretability is somewhat better. This interpretability advantage is because it is possible to visualize the decision tree. The decision tree visualization displays the choices made by the model. This visualization makes it possible for the user of the model to search the database based on the model's selection process and make it possible to achieve a comparable result. However, the process of how the model arrives at the splits is a lot more difficult to understand because this is partly done randomly. The splitting process itself is done by impurity reduction as mentioned in paragraph 3.3.

4.5.4.3 Robustness

Regarding robustness, the decision tree is not very stable. A small change in the data can lead to a large difference in the decision tree. For example, due to a missing or incorrectly entered property of the project for which the price must be predicted and can cause an unreliable outcome. Using the parameter ‘random_state’ in the Python script, it is possible that the splits, of the decision tree, are not made randomly each time the script runs as mentioned in section 3.5. This parameter was added, and the model comes up every run of the Python script with the same results for the same specific case and gives the decision tree some stability, instead of an answer that can be continuously different.

4.5.4.4 Ease of use

The ease of use of this model can be assumed to be relatively good. There is a module available to program the model. Using the Python module results in a programming process in which not every step has to be programmed by the programmer of the model. However, the model takes the projects from the datasheet, which means that there is no input sheet like a Google form that makes it very easy to use the script in a new project. This should be added. The runtime of the script is fast. Within a minute, the results are produced.

4.5.4.5 Boundary conditions

As known from the FALCON model, economic factors, stakeholder requirements, and resource factors are also no parameters in this model. That means that this model is not 100% focused on a specific case, including all the associated factors.

4.6 K nearest neighbors

The last presented method is K nearest neighbors. This paragraph starts with theory implementation in section 4.6.1, thereafter the hyperparameter analysis in section 4.6.2. The numeric results of this method are presented in section 4.6.3. Section 4.6.4 presents the findings of this method.

4.6.1 Theory implementation

A Python script is programmed to execute KNN for cost estimates of bridges.

The KNN model represents a variant of linear regression. To predict the outcome, the model uses the information of neighbouring data points. The outcome is the construction cost of a bridge project.

In the Python model, the predictor of testing and the dependent variables should be defined. In this case, the predictor is the total price of the bridge project, and the dependent variables were equal to the variables in the FALCON model. These variables are, therefore, equal to the input values of the Google form as presented with the FALCON model.
The model uses the same data as used for the other three models. The difference, however, is that this data is normalized before the model uses the data. Using the same database makes it easier to compare the final results of different models. The data is split into a training set and a test set to cover the risk of overfitting.

By the definition of the arguments, the KNN regression is executed. First, the function shuffles the data in random order. Next, it splits the data into test data and training data. The script includes the dimensions of the training data and the test data.

It should be clear that in the case of n-fold cross-validation, the test data consist of 1 project, and the training data consist of 38 projects. The model generates a score for each sample.
The error is calculated by comparing the actual price and the price calculated by the model.

4.6.2 Hyperparameter analysis

The K nearest neighbors' method does have the option to tune parameters. The parameters are a metric parameter, algorithm parameter, leaf size parameter, and the number of neighbors. This section describes all these parameters.

4.6.2.1 Hyperparameter analysis: metric

The metric parameter is related to the distance calculation. Four options are available, Euclidean, Manhattan, Chebyshev, and Minkowski.
The Euclidean distance is the distance between two Cartesian coordinates and is calculated using the Pythagoras theorem (James et al., 2013).
The Manhattan distance is the sum of the absolute differences between corresponding coordinates, as shown in equation 5.

Equation 5: Manhattan distance

$$distance(a,b) = \sum_i |a_i \ \ b_i|$$

The Chebyshev distance is the maximum distance between two vector points. These vector points, for example, x and y, do have standard coordinates xi and yi. This shows equation 6.

Equation 6: Chebyshev distance

$$Chebyshev\ distance\ (x,y) = \max_i(|x_i - y_i|)$$

The Minkowski distance is the generalization of the former distance formulas. X = (x_1, …, x_n) Y = (y_1, …, y_n). If p = 1, the formula is equal to the Manhattan distance. If p=2, the formula is equal to the Euclidean distance. If p goes to infinity, the formula is equal to the Chebyshev distance (Proietti, Panella, Leccese, & Svezia, 2015). Equation 7 shows the Minkowski distance.

Equation 7: Minkowski distance

$$Minkowski\ distance(X,Y) = (\sum_{i=0}^{n} |x_i - \ y_i|^p)^{\frac{1}{p}}$$

4.6.2.2 Hyperparameter analysis: algorithm

The algorithm parameter will represent some algorithm options. These options are the ball tree, kd tree, and brute force.

A ball tree partitions data points into nested sets of hyperspheres in a multidimensional space. It creates an organized way to find the nearest neighbor (Thomas, Vijayaraghavan, & Emmanuel, 2020). A ball tree uses the entire dataset as the root node. The root node is the first node of a tree, in this node, the first division in subsets takes place. The hyperspheres can be separate from each other or overlap each other. Every hypersphere does have a centroid and a radius. The distance from the centre from the ball to a data point will help decide which ball a data point will belong to. A data point will always be in that ball with the shortest distance to the centroid.

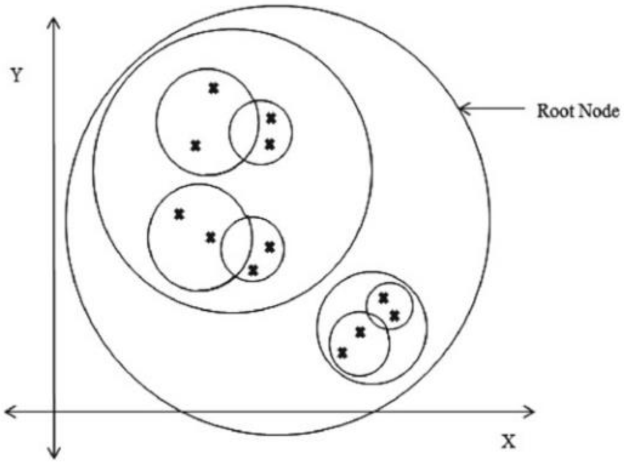

Figure 18: Sample ball tree (Thomas et al., 2020)

To search for the nearest neighbour for a test point t in a ball tree, somewhere in the x-y plane of figure 18; assume that the point p is very close to t, then all sub-trees with child nodes that are further from t than from p are ignored for the rest of the search (Thomas et al., 2020). In case 't' is assumed within the root note circle and close to the smallest circle within the root node, this smallest circle represents a sub-tree and is used for further calculation. The two circles within the smallest circle represent thus a new sub-tree.

Another algorithm is brute force. Brute force is a simpler algorithm, which calculates the distance between the data point of interest to all other data points in the dataset. The classification process assumes that the data point of interest belongs to the class of the majority of its neighbours.

The final algorithm is the KD tree. It will mean K dimensional tree; each node in the tree has K number of nodes.
The construction of a KD tree consists of two essential steps. After traversing the whole tree first step is to find the median and split the dataset along the median. In the case of a two-dimensional dataset, the aim is to build a data structure by picking any random dimension, then the median must be found, and then the dataset splits along this median. The KD tree will search through the data as a binary search tree (Thomas et al., 2020).

Figure 19 shows an example of a KD tree. In this sample, 10 points are shown on different levels of the tree. It is assumed that every point has its own x and y coordinate.

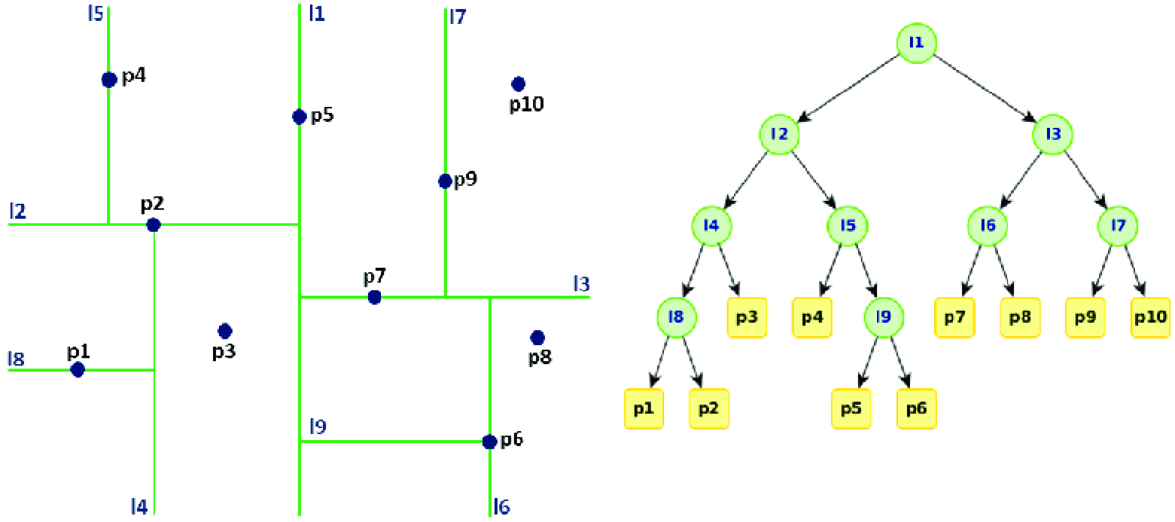

Figure 19: Sample kd tree (Anzola, Pascual, Tarazona, & Gonzalez Crespo, 2018)

4.6.2.3 Hyperparameter analysis: other parameters

The final two parameters are the leaf size and number of neighbors. The leaf size is the number of available projects in the leaf when the tree will switch to another computation method. In the case of a ball tree or a KD tree, the algorithm will change to Brute Force to increase the computation time. The number of neighbors is the number of projects on which the model bases the final output.

4.6.2.4 Hyperparameter analysis: results

The parameters above are considered for defining the optimal set of parameters. In the case of the n-fold hyperparameter analysis, this results in an optimal result for each category. Table 8 shows these results.

Table 8: Results hyperparameter analysis KNN n-fold cross-validation

	Algorithm	Metric	Leaf size	Number of neighbors
Foundation	Ball tree	Chebyshev	18	4
Substructure	Ball tree	Chebyshev	9	1
Superstructure	Ball tree	Chebyshev	9	2
Railing	Ball tree	Chebyshev	1	2

The results of each fold for the hyperparameter optimization are available in Appendix VI.

4.6.3 Results

The next step is the calculation of results for n-fold cross-validation. As mentioned in section 4.3.2, in the case of n-fold cross-validation, the test sample can be compared with the largest number of projects. Therefore, the best results are expected for this type of cross-validation

After the omission of outliers, the boxplots in figure 20 visualize a total of 32 projects. Some values above the boxplot maximum or minimum still occur. Table 9 shows these values. An n-fold cross-validation setting will give the best results. The results are also generated for 5- or 10-fold cross-validation. These results are documented in Appendix VI. The results per project for the n-fold cross-validation and a description of the outliers are also documented in Appendix VI.

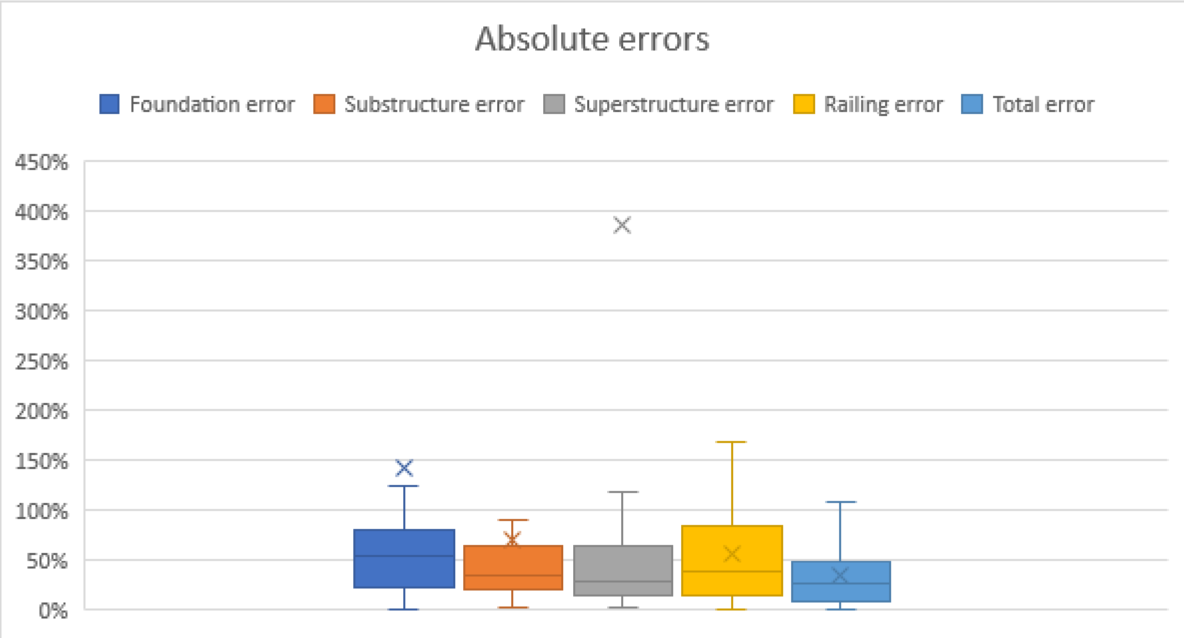

Figure 20: Absolute errors KNN n-fold cross-validation

Table 9: Values boxplots KNN n-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum	2481%	672%			
Value above maximum	363%	395%	10977%		
Value above maximum	303%	147%	278%	306%	
Maximum error	125%	92%	119%	169%	109%
Upper Quartile	82%	64%	64%	84%	49%
Average error 'X'	144%	72%	387%	57%	34%
Median	55%	34%	29%	39%	27%
Lower Quartile	23%	21%	15%	14%	8%
Minimum error	0%	2%	3%	0%	1%

4.6.4 Findings

Based on the results presented in sections 4.6.2 and 4.6.3, this section elaborates on the findings using performance measures as described in section 3.7.3. These performance measures are the accuracy of the predicted total price, described in section 4.6.4.1, the accuracy of the predicted price for components, described in section 4.6.4.2, the interpretability of the results, described in section 4.6.4.3, the robustness of the FALCON model, described in section 4.6.4.4, and the ease of use, described in section 4.6.4.5. Besides some boundary conditions have been described in section 4.6.4.6.

4.6.4.1 Accuracy of the total price

The figures and tables in sections 4.6.2 and 4.6.3 showed the results, including the partial results of each of the four categories. As shown in table 9, the absolute average error on the total price is 34% in the case of n-fold cross-validation. The average error is in the same range as the errors in case the estimate is made by a conventional method. The values are in that case between 15-50% as presented in chapter 1.

The results in Appendix IV showed the average errors of the other cross-validation types. The average error for 10-fold cross validation is 29%, in case of 5-fold cross validation is the average error 41%.

A remarkable value in of course that the predictions in the case of n-fold cross-validation with KNN are less good than in the case of 10-fold cross-validation. A cause that the KNN predictions are not better in case of n-fold cross-validation cannot be immediately indicated.
The following factors play a role in this:

- The results of the partial analyses can balance each other.
- With the hyperparameter analysis and result determination in case of the 10-fold cross-validation, there is a smaller training data set, this can ensure that projects that are together in the test data cannot negatively influence each other in the model while this in case of n-fold cross-validation is still possible.
- Due to the difference in data, the algorithms can find other patterns that may be more favourable for the 10-fold cross-validation variant.
- Outliers in the ‘k’ neighbors can influence the results, as mentioned in paragraph 3.5.

4.6.4.2 Accuracy components

In addition to the accuracy of the model on total price, it is also essential to have insight into the parts' figures as presented. The total price errors are not as large as the error average error of one of the four categories (foundation, substructure, superstructure, railing) would be. The errors of the estimates for one of the categories are for most estimates 10-20% larger than the errors of the estimates for the total price of a project.

4.6.4.3 Interpret the results

The interpretation of the results is in the case of KNN not be that easy since it can switch the calculation method during the execution of the model. Besides, KNN does not indicate which reference projects the model includes in the calculation. The KNN model result is a number that is not substantiated with data and reduces confidence in the data's reliability. It also becomes more challenging to use the model for a project without substantiation of results.

4.6.4.4 Robustness

Robustness must also be considered. Concerning the KNN model, robustness is related to errors and how to deal with them. Errors can be in the model's input, although they can occur in the execution of the model. Section 4.6.4.1 showed the average errors of the other types of cross-validation. From these values it is concluded that the size of the dataset has a significant influence on the performance of this model.

Concerning the input data, the datasheets before the input were defined and are mostly generated by a computer program; the number of errors should therefore be negligible. The next step, from sheet to a category, is done by hand, the chance of errors increases. Although extensive controls after this phase have kept chis chance to a minimum.

4.6.4.5 Ease of use

The ease of use of this model can be assumed to be relatively good. There is a module available to program the model. Using the Python module results in a programming process in which not every step has to be programmed by the programmer of the model. However, the model takes the projects from the datasheet, which means that there is no input sheet like a Google form that makes it very easy to use the script in a new project. This should be added. The runtime of the script is fast. Within a minute, the results are produced.

4.6.4.6 Boundary conditions

As known from the FALCON model, economic factors, stakeholder requirements, and resource factors are also no parameters in this model. That means that this model is not 100% focused on a specific case, including all the associated factors.

4.7 Verification of solutions

In this chapter, the implementation and results of four models, FALCON, KNN, decision tree, and multiple linear regression, are shown. This section verifies the solutions against the requirements. This is part of the research strategy as presented in paragraph 1.4.

In paragraph 2.1 was mentioned that in order to achieve a major performance improvement in cost estimates of bridges, the best performing model that is suitable to make quickly the most accurate cost estimates for bridge projects is needed.

From the statement it follows that two criteria are important regarding the performance. First, relating the computation time, there are no significant differences between the models. All four models can calculate within a couple of minutes an estimate for a bridge project.

The second measure is the accuracy. Table 10 gives an overview of all models and the average errors on the total price of a bridge.

Table 10: Average error predictions total price

	KNN	Decision tree	Multiple linear regression	FALCON
5-Fold	41%	36%	68%	30%
10-Fold	29%	35%	67%	29%
N-fold	34%	36%	57%	24%

In the table above, it became clear that FALCON performs better than all other models in case 5-fold or n-fold cross-validation is applied. Referring back to section 3.5., The results of the multiple linear regression model are, as expected, less good than the other models. In the case of 10-fold cross-validation, the average errors are equal for FALCON and KNN. It shows that the performance of KNN has almost the same accuracy as FALCON. The main difference is that FALCON performs well in all cases.

Regarding interpretability, in the case of FALCON there are no ambiguities as stated in section 4.3.4.3, in the case of KNN there are some ambiguities as stated in section 4.6.4.3. In the case of KNN, the algorithm can be switched during the calculation of a result.

Besides, the boundary conditions are equal for FALCON and KNN. Regarding the ease of use, FALCON has the advantage that new projects are easy calculated by the FALCON model due to the availability of the Google form in the model. In addition, related to the robustness, as shown from the results, the sample size has some influence on the results of FALCON, however the influence is also present for KNN.

To conclude, from these four models, FALCON is the best performing model that is suitable to make quickly accurate cost estimates for bridge projects. This model is validated in the validation phase.

5 Validation

In chapter 4, 4 models including FALCON have been presented for making a cost estimate for bridges. In paragraph 4.7, it became clear that the FALCON model presented the best solution for the construction requirement, as presented in chapter 1.

The aim of this chapter is to check if the FALCON model meets the needs as presented in the research strategy of chapter 1. As mentioned in chapter 2, the first need is that the required calculation time must be reduced from hours to minutes. The second need regards to the accuracy, for Witteveen+Bos, a major step has been taken if the accuracy is improved one level based on the AACE values. This means that the expected deviation on the predicted price with the model may be a maximum of 30%.

At the beginning of the research, the 'needs' became the goal of this research. The best possible solution is presented in Chapter 4. This chapter explains the relationship between the solution offered in Chapter 4 and the requirements that cost estimators themselves set for a cost model.

Therefore, it is essential what the experts' opinion of the model is. Interviews have been organized to evaluate whether the model meets the wishes of the customer. The cost estimators from the Witteveen+Bos company are the customers for this product, the FALCON model.

In total, seven cost estimators have been interviewed. The interviews consist of open and closed questions. The respondents' experience varies from a few years to decades of experience in making cost estimates of civil projects, including road bridges. Therefore, the respondents are a representative group of cost engineers.

The results of the interviews are aggregated and presented in this chapter. The list of 10 main questions, based on the realized model and the obtained results, is available in Appendix VII. The interviews are problem-centred interviews that create the possibility to compare the obtained data with reality. The interviews highlight the individual perspective and do not use a general set of questions (Döringer, 2020).

The questions of the interviews aim to map whether the performance of the FALCON model meets the requirements, the 'needs', of the interviewees and whether it is likely to be implemented in practice by the respondents. Besides, these interviews identify whether the experts have any suggestions for improvement of the realized model.

At the end of this chapter it can then be concluded whether the 'needs' as formulated at the beginning of this study are the same as the 'needs' of the interviewees. In addition, it can be concluded whether the results obtained with the FALCON model meet the 'needs' of the interviewees.

Paragraph 5.1 shows the results of the interviews. In paragraph 5.2, the realized improvements in the FALCON model based on suggestions of the interviewees are described. Subsequently, paragraph 5.3 presents all findings that are based on the interviews.

5.1 Results interviews

This paragraph presents the results of the interviews. In section 5.1.1, the use of reference projects is discussed. Thereafter in section 5.1.2, the collection process of reference projects is discussed. Section 5.1.3 presents if cost estimators have the opinion that models like FALCON can be applied in cost engineering. Subsequently, the expected accuracy for such models is presented in section 5.1.4. Section 5.1.5 presents if the model is suitable for practice according to the interviewees. Improvements are suggested in section 5.1.6 and section 5.1.7. Strengths and weaknesses have been appointed in section 5.1.8 and section 5.1.9. Finally, a couple of other remarks have been listed in section 5.1.10.

5.1.1 Use of reference projects

The results of the FALCON model are based on references. It is interesting to determine whether references are included in the current methods to make a cost estimate of a bridge project.

Three cost estimators have indicated that they are using reference projects already at the start of a project. One of these three cost estimators do not select reference projects at a high level of detail. Another three cost estimators are using reference projects for specific parts of a project to control themselves. One cost estimator remarks that he uses a full reference project if the client requires a copy of that project. One cost estimator mentioned that he does not use reference projects very often. This implicates that in the current work of cost estimators there is not a guideline for using reference projects. Figure 21 shows an overview of these results.

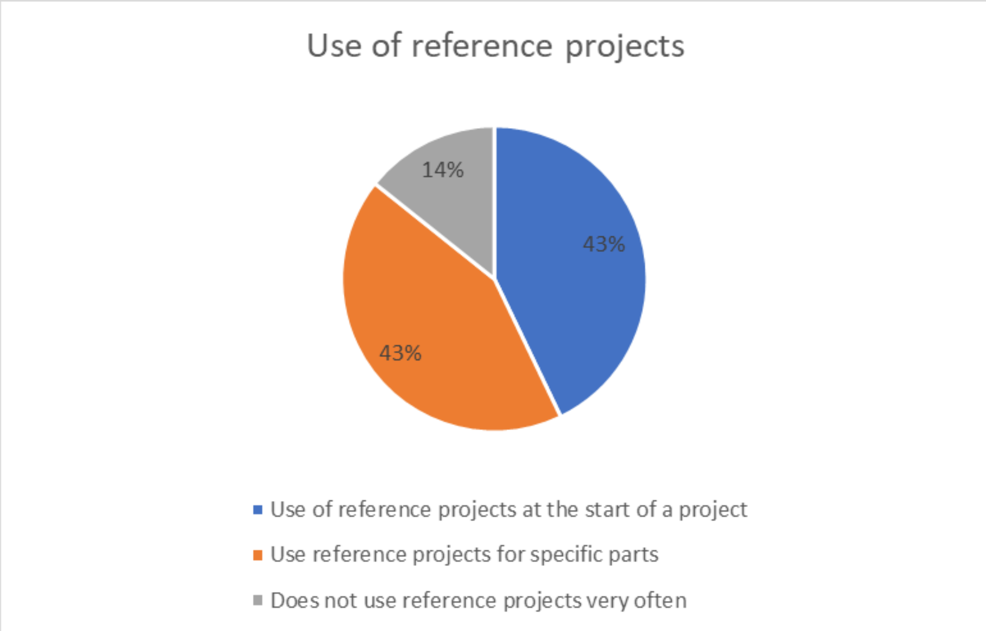

Figure 21: Use of reference projects

5.1.2 Collection of reference projects

The FALCON model uses a database of bridges to estimate the costs of new bridges. Within Witteveen+Bos such a database also exists. Four interviewees make use of this database, but they mention that it does not always work very well. In case it does not work very well, they call a colleague to discuss and/or ask for reference projects. Other interviewees indicate that calling a colleague is their first step in collecting reference projects. This shows that time is one of the important factors in making a cost estimate. Calling colleagues is a solution to speed up the process. It does mean that another colleague loses time making cost estimates for which he or she is responsible. Figure 22 shows an overview of these results.

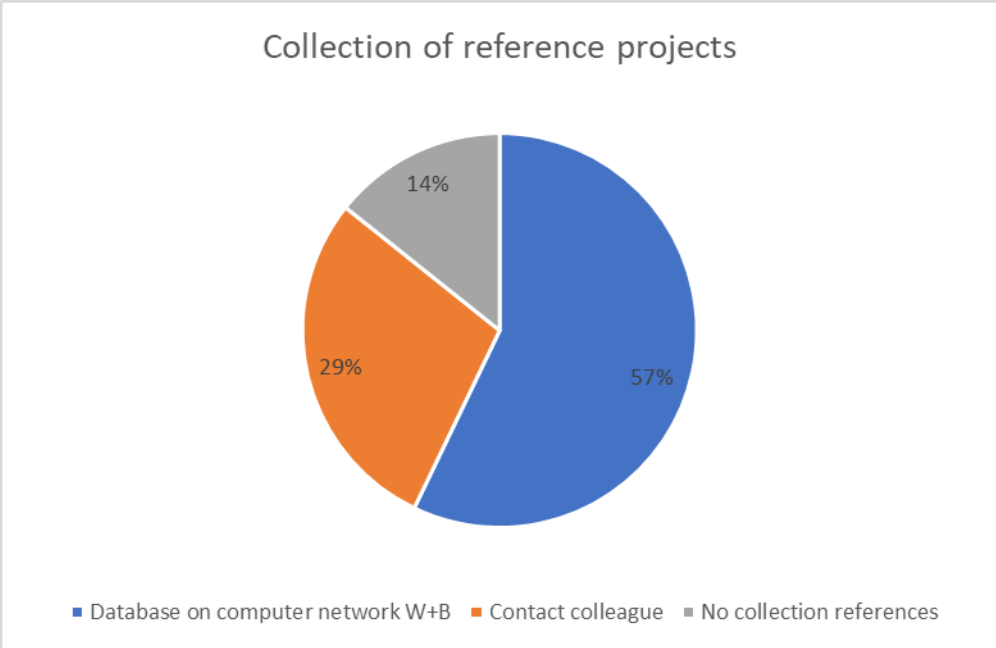

Figure 22: Collection of reference projects

5.1.3 Opportunities for FALCON and Artificial Intelligence

Successful implementation of a FALCON model is only possible if the organization also realizes that such models can also contribute positively to the activities carried out.

All cost estimators are convinced that AI models, like FALCON, can help in cost estimates. Especially in the first stage, with little information available, the interviewees are enthusiastic about the applicability of a FALCON model. Some cost estimators do see opportunities for application in every stage of a project and in case cost estimates for a couple of projects should be made to compare them with each other in the first phase. An overview of these results is shown in figure 23.

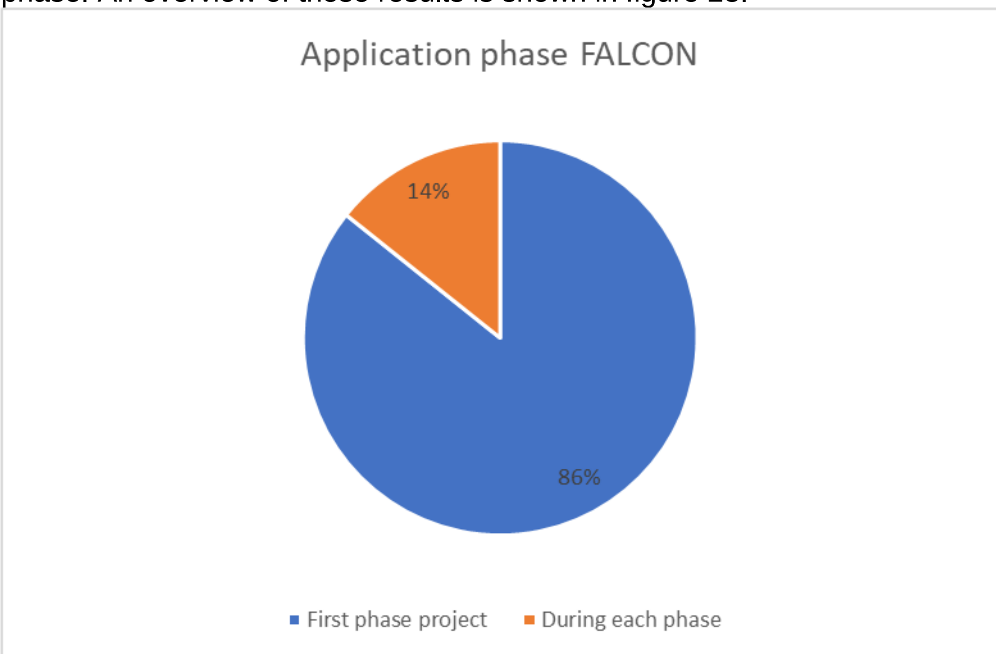

Figure 23: Application phase FALCON

5.1.4 Accuracy

It is crucial which accuracy level is expected. The expected accuracy level can partly determine whether the model FALCON is likely to be used in practice.

Not all cost estimators expect the same accuracy for the total price of a (bridge) project. The estimation deviation from the FALCON model that is accepted differs per cost estimator. One mentioned 40%, another one 30-50%. There is also a cost estimator who is satisfied with an estimation deviation of 15-30%. Another cost estimator expects an estimation deviation of 50% at the first rough estimate, 40% after the first initiative, and it would be nice if 30% estimation deviation is possible. There is also someone who expects an estimation deviation of 40% after the first phase and 25% accuracy after the second phase. The seventh cost estimator expects an estimation deviation of the model of 30%. In sum, the respondents, cost estimators, expect an estimation deviation between 25-40%.

5.1.5 Use in practice

When all conditions are met for a successful implementation of the FALCON model, then the implementation only needs to take place.
All cost estimators were enthusiastic about the model and are expecting to use the model in practice.

5.1.6 Suggested improvements for the FALCON model

In response to the presented model, the respondents suggest some improvements. These improvements are listed and presented in this section.

They advise to add some parameters:

- Span
- Width
- Abutment height
- Architecture superstructure
- Length / width ratio
- Traffic class
- Hydraulic profile

Besides the advice to add parameters, other points of improvement are:
1. Generate a compact input sheet
2. Link quotations to the model.
3. Add more parameters to generate cost estimates for small parts of a bridge project.
4. Create the difference between expensive and cheap projects. A definition will then have to be determined for cheap and expensive.
5. Analyse the other bids of other companies for the same project. As a result of this analysis, more information about a company's bid price can be generated. Is the price for the project fair?
6. Add mandatory fields, for example, the bridge type, amount of supports, and the bridge deck in square meters.
7. Add information about the project. Is the project a new bridge or the replacement of a bridge?

5.1.7 Suggested improvements output FALCON model

The FALCON model can estimate the total price and the estimated price for each of the four parts. However, the cost estimators like to have more output to for example compare projects on their own as an extra check. This is much easier when references are part of the output, visualized with pictures. It would be appreciated if a top 3 of best references can be added to the output.

5.1.8 Strengths

Besides the improvement points, strong arguments are mentioned that can be of added value.

The model gives an estimate based on a reference project. In that case, a contractor has fewer arguments why his product is more expensive if all reference projects have a lower price with the same specifications. The FALCON model has the capacity of being a tool to check our methods, which are used nowadays.

In addition, the model can quickly provide a cost estimate. This certainly adds value if an estimate is requested of a project that should have been completed yesterday, so to speak. Speed is an important factor in these projects.

5.1.9 Weaknesses

There are cases when the added value of the model is negligible. These situations are identified.

The new model is less suitable for cases where a couple of almost the same projects should be compared. These differences are in more detail than the parameters of the Google form of the FALCON model. Another case in which the model is not suitable according to the cost estimators is in case the project a 'one-of-a-kind' project.

5.1.10 Other remarks

The model gives an estimate based on a reference project. In that case, a contractor has fewer arguments why his product is more expensive if all reference projects have a lower price with the same specifications. The FALCON model has the capacity of being a tool to check our methods, which are used nowadays.

Finally, there are comments made by the respondents that do not directly fit the subjects above.

The cost estimators do not agree with each other on the influence of the parameter 'location' on the total price. Someone remarks that all projects should be in the same range. Therefore, it was ok that some projects were not in the database, for example, a highway bridge. It is remarked that the samples' scores give much information, and he also advised to add estimates of him and his colleagues to the model.
The application of new techniques is a point that is considered. Therefore, the model should also be explainable in ordinary language. In the future, the relation between construction cost and investment cost should be made. Just as the generation of key figures to make projects comparable according to the ICMS (International Construction Measurement Standards) standard.

5.2 Implementation results

In addition to the positive comments, points for improvement were suggested. However, the FALCON model is developed in such a way that some of these improvements can be implemented quickly. This paragraph discusses the improvements to the FALCON model.

Further explanation remarks parameters related to improvement

The parameters which should be added, are listed in paragraph 4.2, according to the respondents cannot be easily derived from the available data. Suppose the railing length is set equal to the span of the bridge deck, the data's accuracy in the database decreases. Thus, the parameters span, width, and length/width ratio are not suitable for adding to the database. However, this data is still retrievable for recent projects by requesting additional information, but this is too time consuming for many projects. This argument also applies to the following parameters: abutment height and the hydraulic profile, including width and height. The traffic class is mentioned by the function of the bridge and the scope of the bridge projects. The superstructure architecture is not added as a parameter because the architecture definition is not equal for everyone, which can cause inaccuracy of the data. In this section, several limiting factors have been mentioned. However, these limitations can also serve as the start of new research. In that case, these mentioned limitations have the potential to be removed.

Further explanation of other remarks related to improvement

It was also mentioned to add quotations to the model. However, this would mean new data research. This new approach can undoubtedly be incorporated in a future improvement phase of the model.
Currently, the FALCON model incorporates the lowest bid. It is remarked that an analysis of the other bids of a project could also be useful. In this project, the choice for the lowest bid as input value is a boundary condition. These prices must be estimated and not the other offer prices for a project.

Since the model aims to make an estimate for the early phase of the project, when little information is known, the remark to add more parameters to generate cost estimates for small parts of the bridge is contradictory. Increasing the level of detail is not yet directly relevant in that phase.

Differentiation between expensive and cheap projects can be made. At this stage, when the database consists of a relatively small number of projects, this extra parameter is not directly contributing to a better result. This parameter may be added later after a clear definition of cheap and expensive has been established.

In case it would be possible, the Python model as programmed does not have that option, creating mandatory fields for the input had been implemented. Another suggestion was the remark about the extra project information. In case bridge replacement projects are included in the database, the remark to distinguish between new bridges and bridge replacements can be inserted as a parameter.

For a top 3 of best references, the model must be adjusted in such a way that it can be useful to include this remark in the next improvement stage of this model.

Improvements after interviews

One remark that is made relates to the compactness of the set of entry fields. Although it is simple to enter true or false, a more compact sheet is desired, and the remark is fulfilled. The number of entry fields has decreased from 47 to 30 through a different format of the 'Google' form. Among other things, there is now a dropdown menu for the bridge type. In the first version, true or false had to be entered for each bridge type.

It is remarked that the name of a reference project is useful. In case the model shows a picture of the reference bridge, then an extra improvement has been realized. This comment is processed, and the reference project is now also published in the model's output. Displaying an image is not (yet) possible in this script.
A comparison with the database's mean value is also noted to be added as an extra option. It can give a first indication if a project is expensive or not. This comparison is added as an extra output value.

5.3 Findings

As mentioned in the introduction, this chapter aims to check if the FALCON model meets the needs as presented in the research strategy of chapter 2.

Two needs were defined at the start of this research project. The first need is that the required calculation time must be reduced from hours to minutes. The second need regards to the accuracy, for Witteveen+Bos, a major step has been taken if the accuracy is improved one level based on the AACE values. This means that the expected deviation on the predicted price with the model may be a maximum of 30%.

The summary of the interviews showed the requirements that cost estimators themselves set for a cost model. From the summary of the interviews it became clear that accuracy and computation time are factors that are important according to the cost estimators of Witteveen+Bos.

First, some cost estimators are using phone calls to speed up the search process for reference projects. The second point which relates to the speed is the fact that the speed is mentioned as strong point in the interviews. The fact that the speed is mentioned as one of the strengths and not as an improvement point indicates that the calculation speed of the model meets the expectations of the cost estimators. Besides, the ‘need’, which was determined in the first chapter of this research, to reduce the calculation time from hours to minutes is met. In sum, the predetermined goal regarding the calculation time of a cost estimate, which is achieved with the FALCON model, also meets the expectations from practice.

The second ‘need’ which was predefined at the start of the research regards the accuracy. The FALCON model's accuracy, an average error on the total price of 24%, is in line with what the respondents expect. In general, the interviewees expect a that the predictions of the model deviate 25-40% from the real value. The realised average deviation of the estimations is in line with target, the ‘need’, that is set in chapter 1 at the start of this research. In sum, the average error of the realized estimates with the FALCON model meets predetermined goal regarding the accuracy of a cost estimate, also meets the expectations from practice. The average deviation of the realised cost estimates meets the expectations for the average deviation of the estimations from the respondents, 25-40%, and the target of a maximum deviation of a prediction, which is 30%. However, as shown in table 4 in paragraph 4.3, varies the accuracy of the individual predictions of the FALCON model. In table 4 is shown that 75% of the predictions has a deviation up to 39%, and 25% of the predictions has a deviation between 39% and 75%. These values roughly correspond to the accuracy of cost estimates in the conceptual phase made with a conventional method such as SSK. The deviation from those estimates can be about 50% as presented in chapter 1. Thus, the target of 30% is not met in several cases. Although, more detailed information is available, in the budget authorization or control phase, at the time when the maximum deviation of 30% is normally expected.

Besides, the cost estimators are willing to implement new models, and they see the application possibilities. The interviewees are willing to implement the model in their daily activities. This is also a factor that indicates that the realized model meets the expectations of the interviewees.

In sum, the 'need' regarding the calculation time of the estimate, determined at the start of this project, has been met during this research and this 'need' is in accordance with the interviewees' 'need'. In addition, the 'need' regarding the accuracy of the estimate, determined at the start of this project, has not been met during this research, although the accuracy of the predictions is in accordance with the interviewees' 'need'. Besides, many suggestions of the interviewees are incorporated in the model.

6 Conclusion, discussion, and recommendations

This chapter focuses on the aim of the research, answering the main question. The first paragraph answers the main question. This paragraph presents the conclusion of the research. Thereafter, in paragraph 6.2, the research project is discussed. The last paragraph focuses on future research, and presents some recommendations.

6.1 Conclusion

This chapter presents a conclusion after having taken research steps and leads to the answer to the main question that started this research:

How can FALCON improve the accuracy of cost estimates for road bridge projects in the conceptual phase?

This research presented FALCON as a model that has the potential to be suitable for cost estimates in the conceptual phase of road bridge projects. FALCON is a model in the field of artificial intelligence. Therefore, a model as FALCON is needs to be programmed to use the model in order to solve an optimization problem as in this research project. The first step in order to check if FALCON can improve the accuracy of cost estimates for bridge projects is a programmed version of FALCON. The model must be suitable for the purpose for which it will be used. This means that the model must be suitable for processing of cost data of bridge projects. Thereafter, the realized model must be used to calculate results. Subsequently, conclusions can be drawn based on the results. The modelling step of the FALCON model took place as presented in this research project.

Based on the results calculated with the model, conclusions are drawn. This research has shown that within the field of artificial intelligence, there are various machine learning models that can be used in the field of cost estimates for civil projects. First, this research shows that FALCON can provide more accurate cost estimates for this problem compared to the standard models. In this research, the standard models K nearest neighbors, the decision tree, and multiple linear regression were elaborated for the optimization problem of cost estimates for bridges in the conceptual phase.

The FALCON model offers obviously the best solution for the optimization problem compared to three standard model. The solution offered by the FALCON model must also be substantiated by the existing situation that was outlined prior to the actual research. This comparison shows that the FALCON model can make a fast calculation of the total price of a bridge project in the conceptual phase. The improvement is enormous because the time investment for a cost estimate for bridges in the conceptual phase has been reduced from hours to minutes. This is in accordance with the objective set at the start of the research project.

Calculation time is one of the factors that contributes to a solution that meets the requirements. The most important requirement for a cost estimate is of course the accuracy of the estimate made by the FALCON model. The model can make accurate cost estimates for bridges in the conceptual phase. The accuracy of the estimates is also in accordance with the requirements set by cost estimators in practice, as shown by the interviews with cost estimators from Witteveen+Bos.

More accurate cost estimates are expected in the budget authorization or control phase than in the conceptual phase of a project. The model is not yet suitable for these phases, because the individual results are not all that accurate to meet those requirements. Such an improvement was one of the objectives at the start of the research project.

Referring to the literature, the accuracy of cost estimates in the conceptual phase made by Witteveen+Bos is in line with the standard of the Association for the Advancement of Cost Engineering, AACE. According to those values, the errors of cost estimates in the conceptual are in the range of 15-50%. The cost estimates of bridges in the conceptual phase have an average error of 24% and the individual results also correspond for the great majority to the bandwidth of the AACE. This means that a couple projects had an estimate that deviated more than 50% but less than 75%.

Another major advantage is easiness of explaining the FALCON model. In addition, the employees of Witteveen+Bos see possibilities and are willing to apply the FALCON model in practice. This will benefit to a successful implementation of the FALCON model for cost estimates of bridges.

In sum, the FALCON model is able to calculate a cost estimate, of bridge projects in the conceptual phase, more quickly, and with a comparable accuracy level as with conventional methods that are used today.

6.2 Discussion

In paragraph 6.1, a conclusion is presented for the main question of this research. Applying models like FALCON in the early cost estimates of road bridges is a topic that is constantly evolving. Processing and adopting a technique as presented in papers and applying it to a new case is done in this research project. In this chapter, the conditions related to the design of the model, the data, and the applicability in daily practice are discussed.

Cost data

The cost data have been obtained from the database of Witteveen+Bos. The process from zero to a complete dataset was intensive. Given the number of projects that were eventually found in the dataset, a disproportionate amount of time was spent on the search process, since the obtained dataset is not that large. Besides, the amount of 39 projects not that large if this amount of bridges is compared with the number of bridges that are realized in the Netherlands.
When dividing the data into categories, many sub-parts have been combined into a category. This may mean that a classification has been chosen that, despite the good results, may not have been the most logical choice since it was a time-consuming process.
When processing the results of the interviews, it was also explained that certain parameters were not used because the information about them could not be obtained from the data or with insufficient certainty.
The estimated price by the model is the construction cost tendered by the contractor. However, this is not the price that the contractor ultimately pays for his project. Unfortunately, these prices do not remain transparent to an engineering firm such as Witteveen+Bos to this day. This means that there may be a difference between the estimated price for a project and the actual price of that project.

FALCON model

The quality of the output of a model as FALCON depends mostly on two factors. The model itself, and the quality of the dataset. When the quality of the dataset is low then the predictions are probably less accurate. In this model, it means that the quality of the dataset is less for the data points that are at the edge of the dataset. This means that less comparable projects are close to these projects when all projects are mapped. If the data were distributed more homogeneously, the prediction for those data points at the edge of the set also improves. This is of course also the case when the dataset is expanded with qualitative data.

The validation interviews about the results together with cost estimators were useful. The interviews gave additional insights that are partially implemented. When the model is used, it is recommended that the outcomes that the model generates are regularly checked by a cost estimator as a double-check. The approval of an expert gives extra security to the model output in addition to the results of the model which are already presented.

Concerning the predicted results, this model does not make predictions using a feedback system. The FALCON model is not a self-learning model. Improvement or optimization steps must therefore mainly be carried out by the users themselves.

The output can also be optimized using other models like fast messy genetic algorithm. That FALCON can be used together with fast messy genetic algorithm is already mentioned in paragraph 2.1.

Due to the main question, the model is now mainly assessed on one property. However, this does not mean that no other benefits have been achieved in addition to the improvement of cost estimates for road bridge projects in the early phase of a project. One of these benefits is that the model can also be used when looking for references.

Optimum solution
In this project, the improvement for cost estimates for road bridge projects has been investigated. In this research project, 4 different models have been presented. However, it is not known if FALCON is the best solution to this problem. For example, it could be possible that another work breakdown structure will increase the performance of the FALCON model or one of the models as presented in chapter 4.

Comparability projects
The results produced by the FALCON model depend on several factors. Given the number of different options that can be chosen, one may wonder whether this does not lead to too many different data points. Said differently; the variation in projects may be found too large for the number of projects in the database. This could be one of the reasons why the deviation in the estimated total price is larger than in the paper by Wang et al. (2017) on which this research is partly based, in that case only concrete multi-story buildings were investigated.

6.3 Recommendations
This paragraph is one with a vision for the future. After all, there are still many opportunities for research in the field of cost management and artificial intelligence after this research. A couple of recommendations are given for future research.

Data
Before any type of artificial intelligence is applied, the available data set must be properly mapped out. This argument applies to any type of model that depends on historical data. If the available data is mapped out in advance, it becomes much clearer what the opportunities are. The researcher makes a well-considered choice for the research method in that case.

It must also be considered that the available data often needs to be processed or adjusted to be suitable for a computer model as FALCON. This can be a time-consuming process. In advance of a new research project, it must be mapped out if many data processing steps are required to use the data in the new model.

Optimize output FALCON
The output of FALCON has the potential to be improved. Improvement of the results should be realized by using other models. One model that can be considered in that case is fmGA (fast messy genetic algorithm). In paragraph 2.1, it is already presented that FALCON can be used in combination with fmGA. The advantage of this method is that the predicted results of FALCON can be improved by an extra iteration. In case a completely different model is chosen, one must start again from scratch.

Other models
As mentioned in the report. FALCON is not a trained artificial intelligence model. Other artificial intelligence models are also applicable to this problem. These models could probably decrease the deviation between an estimate and the real value of a project. In the final comparison, it becomes clear which model is most suitable to apply to this issue. Besides, by use of a trained model, a comparison can be made between untrained and trained models.

Other project types

The FALCON model has the potential to be applied in other areas within the infrastructure. Examples of these areas are dikes, highways, and quay walls. In case the dataset size is somewhat equal to this project, it is recommended to choose for less complex projects. Although the 4 cost categories used in this project will not all be applicable in the new situation, it is of course possible to define new categories following the same approach as in this project through an analysis of the new cost data.

Wide application FALCON

This project was based on the Dutch RAW standard and the index numbers of the Dutch organization called CBS are used. This is not a problem for the implementation of this model in other companies in the Netherlands. Although, for international implementation, the use of international standards like the International Construction Management Standard would be useful. Implementation of that standard to a new project is recommended.

Bibliography

Abraham, A. (2004). Neuro Fuzzy Systems: Sate-of-the-Art Modeling Techniques. Retrieved from https://arxiv.org/abs/cs/0405011

Ahiaga-Dagbui, D. D., & Smith, S. D. (2014). Rethinking construction cost overruns: cognition, learning and estimation. *Journal of Financial Management of Property and Construction*.

An, S.-H., Park, U.-Y., Kang, K.-I., Cho, M.-Y., & Cho, H.-H. (2007). Application of support vector machines in assessing conceptual cost estimates. *Journal of Computing in Civil Engineering, 21*(4), 259-264.

Anzola, J., Pascual, J., Tarazona, G., & Gonzalez Crespo, R. (2018). A clustering WSN routing protocol based on kd tree algorithm. *Sensors, 18*(9), 2899.

Badra, I., Badawy, M., & Attabi, M. Conceptual Cost Estimate of Buildings Using Regression Analysis In Egypt.

Bates, J., Burton, C. D. J., Creese, R. C., Hollmann, J. K., Humphreys, K. K., McDonald Jr, D. F., & Miller, C. A. (2005). Cost estimate classification system–as applied in engineering, procurement, and construction for the process industries.

Bouwkostenindex. (2020). Retrieved from http://www.bouwkostenindex.nl/Index/Select

Brownlee, J. (2016). *Deep learning with Python: develop deep learning models on Theano and TensorFlow using Keras*: Machine Learning Mastery.

Burghate, M. (2018). Work Breakdown Structure: Simplifying Project Management. *International Journal of Commerce Management, 3*(2).

Burke, R. (1999). Project Management, Planning and Control Techniques. *International Journal of Operations Production Management*

19(12), 1335-1337.

CBS. (2020). Grond-, weg- en waterbouw (GWW); inputprijsindex 2000 = 100, vanaf 1979. Retrieved from https://opendata.cbs.nl/statline/#/CBS/nl/dataset/81139ned/table?ts=1597053344313

Cheng, M.-Y., Tsai, H.-C., & Hsieh, W.-S. (2009). Web-based conceptual cost estimates for construction projects using Evolutionary Fuzzy Neural Inference Model. *Automation in Construction, 18*(2), 164-172.

CROW. (2018). *Standaardsystematiek voor kostenramingen – SSK-2018*. Retrieved from https://kennisbank.crow.nl/kennismodule/detail/34734#34712

Dechoretz, J. A. (Producer). (2011). How to Develop Parametric Cost Estimating Relationships (CER). [Powerpoint slides] Retrieved from https://www.dace.nl/download/?id=14593404

Döringer, S. (2020). 'The problem-centred expert interview'. Combining qualitative interviewing approaches for investigating implicit expert knowledge. *International Journal of Social Research Methodology*, 1-14.

Du Bois, B., Fletcher, W., & Danks, A. (2017). *Guidance Note 2 Base Cost Estimation*. Retrieved from Canberra:

Elmousalami, H. H. J. H. (2019). Intelligent methodology for project conceptual cost prediction. *5*(5), e01625.

Elmousalami, H. H. J. H. (2020). Artificial Intelligence and Parametric Construction Cost Estimate Modeling: State-of-the-Art Review. *Journal of Construction Engineering Management*

146(1), 03119008.

Fushiki, T. (2011). Estimation of prediction error by using K-fold cross-validation. *Statistics and Computing, 21*(2), 137-146.

Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2019). A generalized mean distance-based k-nearest neighbor classifier. *Expert Systems with Applications, 115*, 356-372.

Hatamleh, M. T., Hiyassat, M., Sweis, G. J., & Sweis, R. J. (2018). Factors affecting the accuracy of cost estimate: case of Jordan. *Engineering, Construction and Architectural Management*.

Hong, T., Wang, Z., Luo, X., & Zhang, W. (2020). State-of-the-art on research and applications of machine learning in the building life cycle. *Energy and Buildings, 212*, 109831.

Hsiao, F. Y., Wang, S. H., Wang, W. C., Wen, C. P., & Yu, W. D. (2012). Neuro-fuzzy cost estimation model enhanced by fast messy genetic algorithms for semiconductor hookup construction. *Computer-Aided Civil Infrastructure Engineering*
27(10), 764-781.
Idrus, A., Nuruddin, M. F., & Rohman, M. A. (2011). Development of project cost contingency estimation model using risk analysis and fuzzy expert system. *Expert Systems with Applications, 38*(3), 1501-1508.
James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112): Springer.
Koch, C. (2019). *Machine Learning for Engineering Processes.* Paper presented at the International Conference on Business Information Systems.
Kurnia, R. F., Latief, Y., & Riantini, L. S. (2018). *The Development of Risk-Based Standardized WBS (Work Breakdown Structure) for Cost Estimation of Concrete Precast Bridge Construction.* Paper presented at the 8th International Conference on Industrial Engineering and Operations Management, IEOM 2018.
Laken, v. d., P. (21-10-2020). Neural Networks. Retrieved from https://paulvanderlaken.com/2017/10/16/neural-networks-101/
Lamein, R. (2016). *Publieke opinie EV's en EV Aladdin.* (BSc). Universiteit van Amsterdam, Amsterdam.
Latief, Y., Nurdiani, D., & Supriadi, L. S. R. (2019). *Development of work breakdown structure (WBS) dictionary for the construction works of lower structure steel bridge.* Paper presented at the MATEC Web of Conferences.
Lee, S.-H., Lee, H. J., Park, S. I., Choi, K.-W., & Kwon, T. H. (2016). A PLIB-based New Bridge Breakdown System Considering Functional Properties-Focused on Geometric Modeling. *Journal of the Computational Structural Engineering Institute of Korea, 29*(4), 335-345.
Lemon, S. C., Roy, J., Clark, M. A., Friedmann, P. D., & Rakowski, W. J. A. o. b. m. (2003). Classification and regression tree analysis in public health: methodological review and comparison with logistic regression. *26*(3), 172-181.
Leśniak, A., & Zima, K. (2018). Cost calculation of construction projects including sustainability factors using the Case Based Reasoning (CBR) method. *Sustainability, 10*(5), 1608.
Lim, B., Nepal, M. P., Skitmore, M., & Xiong, B. (2016). Drivers of the accuracy of developers' early stage cost estimates in residential construction. *Journal of Financial Management of Property and Construction*.
Lin, C.-J., & Lin, C.-T. (1997). An ART-based fuzzy adaptive learning control network. *IEEE Transactions on Fuzzy Systems, 5*(4), 477-496.
Lin, C.-T., Lin, C.-J., & Lee, C. G. (1995). Fuzzy adaptive learning control network with on-line neural learning. *Fuzzy sets and systems, 71*(1), 25-45.
Ling, W.-K. (2010). *Nonlinear digital filters: analysis and applications*: Academic Press.
Mahamid, I. (2011). Early cost estimating for road construction projects using multiple regression techniques. *Construction Economics and Building, 11*(4), 87-101.
Nehi, H. M., & Maleki, H. R. (2005). *Intuitionistic fuzzy numbers and it's applications in fuzzy optimization problem.* Paper presented at the Proceedings of the 9th WSEAS International Conference on Systems.
Nikam, S., Nikumbh, P., & Kulkarni, S. (2012). Fuzzy logic and neuro-fuzzy modeling. *International Journal of Computer Applications, 4*, 22-31.
Niu, M., Li, Y., Wang, C., & Han, K. (2018). RFAmyloid: a web server for predicting amyloid proteins. *International Journal of Molecular Sciences, 19*(7), 2071.
Nugroho, A., & Latief, Y. (2018). *Development of Standardized WBS (Work Breakdown Structure) Based on Risk for Cost Control of Steel Bridge Construction Projects.* Paper presented at the 8th International Conference on Industrial Engineering and Operations Management, IEOM 2018.
Oates, T., & Jensen, D. (1997). *The e! ects of training set size on decision tree complexity.* Paper presented at the Proc. 14th Int. Conf. on Machine Learning.

Østergård, T., Jensen, R. L., & Maagaard, S. E. (2018). A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy, 211*, 89-103.
Peterson, L. E. J. S. (2009). K-nearest neighbor. *4*(2), 1883.
Proietti, A., Panella, M., Leccese, F., & Svezia, E. J. M. (2015). Dust detection and analysis in museum environment based on pattern recognition. *66*, 62-72.
Reich, Y. (1997). Machine learning techniques for civil engineering problems. *Computer-Aided Civil Infrastructure Engineering, 12*(4), 295-310.
Rev, E. (2003). *PROJECT MANAGEMENT PRACTICES Work Breakdown Structure*.
Rockafellar, R. T., Uryasev, S., & Zabarankin, M. (2008). Risk tuning with generalized linear regression. *Mathematics of Operations Research, 33*(3), 712-729.
Rodríguez, Y., Falcón, R., Varela, A., & García, M. M. (2008). Learning membership functions for an associative fuzzy neural network. In *Granular Computing: At the Junction of Rough Sets and Fuzzy Sets* (pp. 151-161): Springer.
Roozenburg, N. F. M., & Eekels, J. (1995). *Product Design: Fundamentals and Methods*: Wiley.
Serpell, A. F. (2005). Improving conceptual cost estimating performance. *AACE International Transactions*, ES131.
Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*: Cambridge university press.
Shihabudheen, K., & Pillai, G. (2017). Regularized extreme learning adaptive neuro-fuzzy algorithm for regression and classification. *Knowledge-Based Systems, 127*, 100-113.
Shin, Y. (2015). Application of boosting regression trees to preliminary cost estimation in building construction projects. *Computational intelligence and neuroscience, 2015*.
Strobl, C., Malley, J., & Tutz, G. J. P. m. (2009). An introduction to recursive partitioning: rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *14*(4), 323.
Syed, M. E. (2014). *Attribute weighting in k-nearest neighbor classification*. Retrieved from
Thomas, T., Vijayaraghavan, A. P., & Emmanuel, S. (2020). *Machine Learning Approaches in Cyber Security Analytics*: Springer.
Van der Linde, G. J., Modders, S., & Van Dissel, M. (2017). *Gemeente Wormerland Beheerplan civieltechnische kunstwerken*. Retrieved from Harderwijk:
Wang, W.-C., Bilozerov, T., Dzeng, R.-J., Hsiao, F.-Y., & Wang, K.-C. (2017). Conceptual cost estimations using neuro-fuzzy and multi-factor evaluation methods for building projects. *Journal of Civil Engineering Management*
23(1), 1-14.
Wu, J., Chen, X.-Y., Zhang, H., Xiong, L.-D., Lei, H., & Deng, S.-H. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology, 17*(1), 26-40.
Zadeh, L. A., Yager, R. R., Ovchinnikov, S., Tong, R. M., & Nguyen, H. T. (1987). Outline of a New Approach to the Analysis of Complex Systems and Decision Process.
. *Fuzzy Sets and Applications: Selected Papers by LA Zadeh*, 105-146.
Zhou, L. (2018). A Machine Learning Approach for Conceptual Cost Estimation.

Appendix I

Index data

During this research, two options to index prices have been investigated. Indexing the data is the first step to make cost data of bridges comparable. These two indexes are described in this appendix. First, the 'bouwkostenindex' of the Calcsoft company is described, thereafter the CBS index is described.

Content

1. Bouwkostenindex Calcsoft
2. CBS index

1. Bouwkostenindex Calcsoft

For the cost, data swaps are necessary regarding the index of prices. These swaps are one of the steps to make data comparable. Therefore, an index table regarding road bridges is the most optimal. In the Netherlands, an index for bridges is available. This index is created by Calcsoft BV and available by use of a license of Witteveen+Bos. In the graph below, 1-1-2010 is 100.

Important to note is the difference between the construction cost index (in Dutch: bouwkostenindex) and the tender index (in Dutch: aanbestedingsindex). The construction cost index visualizes the development of the actual construction costs over time. The tender index visualizes the development of the bids made by contractors on projects. The difference between these indexes therefore also suggests the profits within the companies.

In this project, bids of contractors are used as input for the model. The graph in figure 24 indicates how projects in the former 12 years can be placed in time. Table 11 presents the numerical values.

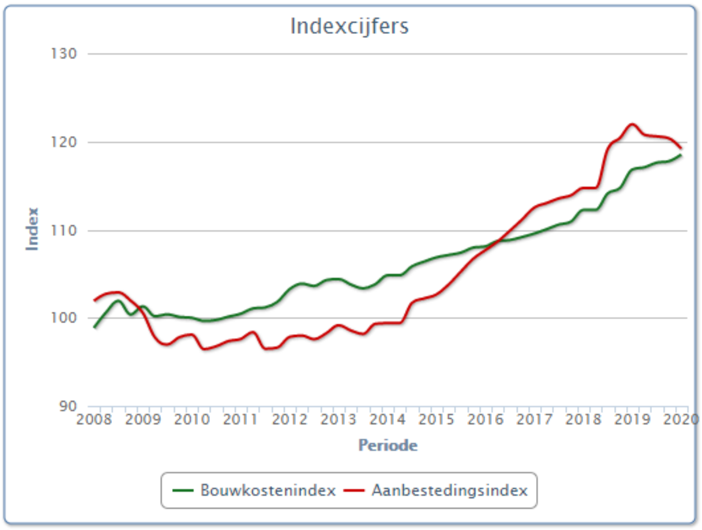

Figure 24 Index bridge price period 2008-2020 ("Bouwkostenindex," 2020)

Table 11 Construction cost index and tender index("Bouwkostenindex," 2020)

	Construction cost index	Percentage difference	Tender index	Percentage difference
Jan-08	98.86		101.91	
Jan-09	101.3	2.47	100.56	-1.32
Jan-10	100	-1.28	98.09	-2.46
Jan-11	100.45	0.45	97.6	-0.50
Jan-12	103.25	2.79	97.83	0.24
Jan-13	104.4	1.11	99.16	1.36
Jan-14	104.81	0.39	99.39	0.23
Jan-15	106.86	1.96	102.65	3.28
Jan-16	108.1	1.16	107.68	4.90
Jan-17	109.55	1.34	112.5	4.48
Jan-18	112.23	2.45	114.73	1.98
Jan-19	116.82	4.09	121.95	6.29
Jan-20	118.53	1.46	119.18	-2.27

2. CBS index

More than one index is available for bridges. The governmental organization CBS (Centraal Bureau voor de Statistiek) presents index numbers for bridges since 2000. This creates an opportunity to also implement projects from the period 2000-2008.

CBS used for all his indexes PPI, DPI, and CPI data. This means Product Price Index, Service Price Index, and Consumer Price Index (in Dutch: Producenten Prijs Index, Diensten Prijs Index, and Consumenten Prijs Index). In other words, prices paid by clients are used.
For this project, the annual indices of CBS are used. This creates the option to also use projects in the period before 2008. Via this method, more data points can be used. The numbers are presented in table 12.

Table 12 CBS index bridges(CBS, 2020)

Measurement moment index	Index number (2000=100)	Percentage difference
Jan-00	98.5	
Jan-01	104.7	6.3
Jan-02	107.1	2.3
Jan-03	111.4	0.8
Jan-04	117.6	-1.3
Jan-05	107.1	0.5
Jan-06	111.4	4
Jan-07	117.6	5.5
Jan-08	122	3.7
Jan-09	130.9	7.3
Jan-10	119	-9.1
Jan-11	123.5	3.7
Jan-12	124.5	0.8
Jan-13	126	1.2
Jan-14	126.4	0.3
Jan-15	124.7	-1.3
Jan-16	124.1	-0.5
Jan-17	128.2	3.2
Jan-18	132.6	3.5
Jan-19	135.3	2
Jan-20	137.3	1.5

Appendix II

RAW descriptions for bridges

In this appendix, the categories of work for bridge projects are mentioned including the RAW descriptions. These RAW descriptions are mentioned per category of work.

1. Preparatory work
10.1 Tijdelijke maatregelen en voorzieningen
10.31.12 Verwijderen hek
11.22 Sloopwerk betonwerk
11.23 Sloopwerk staalwerk
25.11 Verwijderen riolering
34.02.02 Verwijderen lichtmast
47.21 Verwijderen duikers
51.01 Opruimingswerk
51.41 Verwijderen beplantingen
51.51 Verwijderen bomen en stobben
52.71 Verwijderen constructies (beschoeiing)
81.01 - 81.07 Verwijderen verharding

2. Soil work
22.01 Grond ontgraven
22.02 Grond vervoeren
22.03 Grond verwerken
22.04 Grond scheiden, verdichten en profileren
22.1 Cultuurtechnisch grondwerk
52.11 Bestorting als verdediging c.q. filter
52.71 Verwijderen constructies

3. Foundation Structure (including bemaling/kuip/damwanden)
21 Bemalingen
41 Funderingsconstructies
44 Houtconstructies

4. Substructure (pijlers, vloer, landhoofd, vleugelwand, ligger, opleggingen, deksloof, poeren, keerwand)
42.11 Toepassen bekisting
42.11.31 Aanbrengen werkvloer
42.13 Toepassen ondersteuningsconstructie
42.2 Vooraf vervaardigde betonelementen
42.21 Aanbrengen betonstaal
42.31 Aanbrengen beton
42.55 Aanbrengen opleggingen
43.17 Transporteren en monteren staalconstructies
44 Houtconstructies
47.31.21 Aanbrengen betonnen stootplaten
80.03 Aanbrengen van een gebonden wegfundering

5. Superstructure (randen, vloer, ankers (leuning), voegconstructie, schampkanten)
42.11 Toepassen bekisting
42.11.31 Aanbrengen werkvloer
42.13 Toepassen ondersteuningsconstructie
42.21 Aanbrengen betonstaal
42.31 Aanbrengen beton
43.17 Transporteren en monteren staalconstructies
47.31.11 Aanbrengen brugdek van prefab betonelementen
44 Houtconstructies

6. Pavements
32 Wegbebakening
81 Bitumineuze verhardingen
82 Betonverhardingen
83 Elementenverhardingen
42.81 Afwerken taluds (verharding)

7. Railings
33.11 Aanbrengen geleiderailmaterialen
33.41 Aanbrengen betonnen geleidebarrier

8. External finishes (Groeninrichting / afwatering groenvoorzieningen / openbare ruimte)
25 Riolering (aanbrengen)
42.11 Toepassen bekisting
42.11.31 Aanbrengen werkvloer
42.31 Aanbrengen beton
47.22 Duikers
34 Verlichting
10.31 Afrasteringen
51 Groenvoorzieningen

9. General Cost
56 Conserveringswerken
41.xx Maken berekeningen en tekeningen
42.xx Maken berekeningen en tekeningen
43.xx Maken berekeningen en tekeningen
62 Tijdelijke verkeersmaatregelen
41.24 Uitvoeren metingen
10.13 Tijdelijke voorzieningen (hekwerken, gronddepot)
10.23 Communicatie
10.05 Inzetten werknemers en materieel
10.12 Gebruik hulpmiddelen
10.16 Tijdelijke voorzieningen bescherming te handhaven beplanting
18.14 Baggerdepots

10. Overhead
10.11 Werkterrein
92 Uitvoeringskosten
93 Algemene kosten
94 Winst en risico
01.08 Bijdragen
10.03 Stelposten

Appendix III

Extensive explanation results FALCON

This appendix shows all results and predictions of tests with the FALCON model. For the generation of the results, cross-fold validation is used as method. Cross fold validation is possible with different sample sizes. In this research project, results are generated for three different test sample sizes: 1, 4, 8 (n-fold, 5-fold, 10-fold). In the end, it also shows if the sample size has some influence on the results of this method.
Before the results of 10-fold and 5-fold cross-validation are generated is an improvement of the results by a hyperparameter analysis investigated. The second section explains the hyperparameter analysis. The 10-fold cross-validation results are shown in the third section. The fourth section shows the 5-fold cross-validation results.

Content

1. N-fold cross-validation
2. Hyperparameter analysis
3. 10-Fold cross-validation
4. 5-Fold cross-validation
5. Findings

1. N-fold cross-validation

In the case of n-fold cross-validation, the amount of test samples is equal to the number of projects in the database. Table 13 shows the results of all separate projects, in this case, equal to the size of a sample, which is included in the visualization.

Table 13: Absolute errors FALCON N-fold cross-validation

	Total absolute error in %	Foundation absolute error in %	Substructure absolute error in %	Superstructure absolute error in %	Railing absolute error in %
Project 2	42%	71%	35%	18%	203%
Project 3	93%	41%	87%	98%	0%
Project 4	2%	13%	2%	29%	29%
Project 7	50%	250%	34%	61%	61%
Project 8	6%	113%	67%	41%	234%
Project 10	43%	11%	62%	99%	34%
Project 11	1%	107%	24%	38%	100%
Project 12	36%	38%	23%	57%	39%
Project 13	23%	46%	5%	1%	94%
Project 14	2%	7%	2%	18%	23%
Project 15	20%	43%	47%	26%	33%
Project 16	7%	23%	21%	24%	22%
Project 17	46%	61%	31%	61%	17%
Project 18	3%	31%	27%	31%	18%
Project 19	75%	107%	26%	497%	76%
Project 20	14%	75%	55%	52%	179%
Project 21	65%	66%	78%	46%	54%
Project 22	24%	199%	20%	49%	100%
Project 23	5%	62%	28%	5%	83%
Project 24	2%	32%	4%	1%	49%
Project 25	10%	70%	14%	66%	87%
Project 26	40%	52%	85%	83%	64%
Project 27	3%	41%	43%	60%	100%
Project 28	28%	9%	138%	22%	67%
Project 29	9%	54%	416%	31%	28%
Project 30	29%	43%	7%	50%	32%
Project 32	2%	63%	23%	28%	2%
Project 33	6%	39%	19%	40%	2%
Project 34	38%	36%	32%	25%	1%
Project 36	25%	58%	30%	45%	34%
Project 37	2%	77%	30%	47%	9%
Project 38	26%	24%	80%	26%	6%
Project 39	19%	63%	35%	7%	26%

Project 1, 5, 6, 9, 31, and 35 are not included in this list. These projects all form a real deviation from the data. This does not mean that these six projects are bad data points. Currently too little comparable material is available in the dataset. Therefore, it is difficult to make a cost estimate for these projects if they are the test case.
Project 1 is unique because of a lot of ground reinforcement. Project 5 is unique because of the concrete culvert construction. Project 6 is unique because the bridge is movable. Project 31 is unique because it concerns an old bridge that is replaced with a new one. Project 35 is unique because of the size of the bridge deck, 1132m2 is a lot larger than the other projects in the dataset.

2. Hyperparameter analysis

In the hyperparameter analysis, the triangular function is replaced by a Gaussian membership function. After testing all projects, the same projects as mentioned above were omitted and the results of this analysis are shown in table 14. On average the results as shown below are not an improvement in comparison with the results presented in the former section. Therefore, the 5-fold and 10-fold cross-validation is not executed by use of a Gaussian membership function.

Table 14: Results hyperparameter analysis Gaussian membership function

	Foundation error in %	Substructure error in %	Superstructure error in %	Railing error in %	Total absolute error in %
Project 2	71	89	18	62	60
Project 3	41	87	98	0	93
Project 4	13	2	29	46	1
Project 7	250	16	69	57	52
Project 8	90	67	41	9	58
Project 10	0	63	99	58	11
Project 11	107	24	38	100	1
Project 12	34	28	57	91	58
Project 13	46	5	1	113	23
Project 14	7	2	18	32	0
Project 15	43	47	26	9	27
Project 16	23	21	24	8	14
Project 17	61	31	61	7	38
Project 18	31	27	31	9	10
Project 19	107	26	497	27	88
Project 20	96	31	7	130	22
Project 21	29	78	52	20	55
Project 22	87	73	49	85	60
Project 23	62	403	5	37	70
Project 24	58	4	1	53	2
Project 25	71	54	53	100	5
Project 26	52	94	83	5	26
Project 27	41	362	60	0	109
Project 28	9	138	22	38	33
Project 29	54	416	31	0	2
Project 30	43	51	50	111	17
Project 32	63	23	28	30	5
Project 33	39	19	40	23	1
Project 34	5	16	10	35	30
Project 36	58	30	69	20	46
Project 37	77	30	47	61	6
Project 38	418	80	17	21	7
Project 39	63	25	7	34	32
Average	68	75	53	43	32

3. 10-Fold cross-validation

In the case of 10-Fold cross-validation, the test sample size is 4. In each test sample are thus 4 different projects.

In the second table, the results for each project are shown for 10-fold cross-validation. Figure 25, table 15, and table 16 show the results. In the n-fold cross-validation section, some project results were omitted. The omitted results are the same projects as in the case of n-fold cross-validation. The values above a boxplot maximum are only shown in table 15.

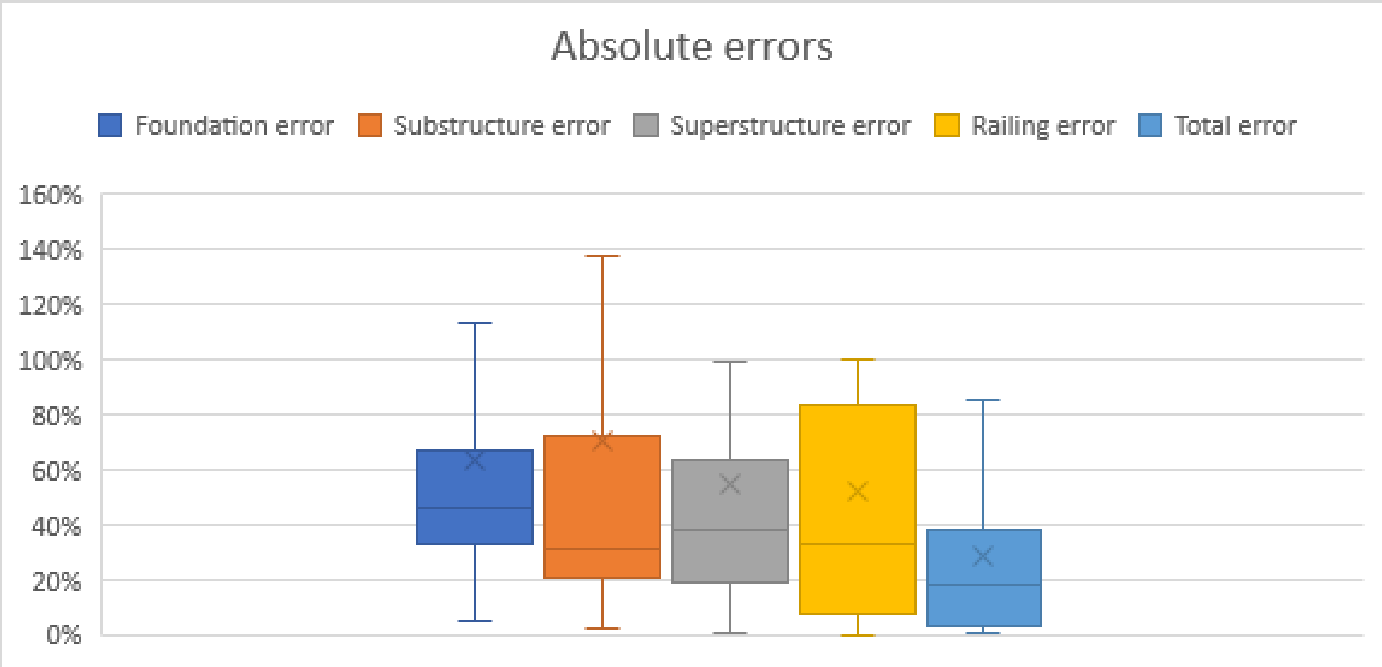

Figure 25: Absolute errors FALCON 10-Fold cross-validation

Table 15: Values Boxplots FALCON 10-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum		416%			108%
Value above maximum		403%		297%	100%
Value above maximum	500%	362%	497%	203%	92%
Maximum error	113%	138%	99%	100%	85%
Upper Quartile	67%	73%	64%	83%	38%
Average error 'X'	64%	70%	55%	52%	29%
Median	46%	31%	38%	33%	18%
Lower Quartile	33%	21%	19%	8%	4%
Minimum error	5%	2%	1%	0%	1%

Table 16: Absolute errors FALCON 10-fold cross-validation

	Absolute error Foundation in %	Absolute error substructure in %	Absolute error superstructure in %	Absolute error railing in %	Absolute total error in %
Project 2	71%	35%	18%	203%	42%
Project 3	41%	87%	98%	0%	93%
Project 4	13%	2%	29%	29%	2%
Project 7	43%	16%	88%	61%	19%
Project 8	113%	67%	41%	6%	13%
Project 10	60%	35%	99%	34%	16%
Project 11	107%	24%	38%	100%	1%
Project 12	34%	23%	57%	39%	47%
Project 13	46%	5%	1%	94%	23%
Project 14	7%	2%	18%	23%	2%
Project 15	43%	47%	26%	33%	20%
Project 16	23%	21%	24%	22%	7%
Project 17	61%	31%	61%	17%	100%
Project 18	31%	27%	31%	18%	3%
Project 19	107%	26%	497%	7%	85%
Project 20	67%	31%	7%	297%	2%
Project 21	29%	78%	29%	50%	37%
Project 22	68%	20%	49%	100%	36%
Project 23	62%	403%	5%	83%	92%
Project 24	53%	4%	1%	100%	3%
Project 25	71%	54%	66%	86%	3%
Project 26	44%	85%	83%	64%	39%
Project 27	41%	362%	60%	0%	108%
Project 28	9%	138%	22%	67%	28%
Project 29	54%	416%	31%	28%	9%
Project 30	43%	26%	50%	32%	23%
Project 32	55%	47%	20%	1%	5%
Project 33	5%	19%	12%	0%	4%
Project 34	36%	32%	82%	1%	11%
Project 36	58%	4%	45%	34%	18%
Project 37	77%	30%	47%	9%	2%
Project 38	500%	80%	75%	6%	31%
Project 39	28%	35%	7%	84%	18%

4. 5-Fold cross-validation

In the case of 5-Fold cross-validation, the test sample size is 8. In each test sample are thus 8 different projects.

In the second table, the results for each project are shown for 5-fold cross-validation. Figure 26, table 17, and table 18 show the results. In the n-fold cross-validation section, some project results were omitted. The omitted results are somewhat the same projects as in the case for n-fold cross-validation. In this case, one extra project is omitted. The prediction for the foundation of project 12 becomes bad because the most comparable project for the foundation of project 12 was inside the test sample. The values above a boxplot maximum are only shown in table 17.

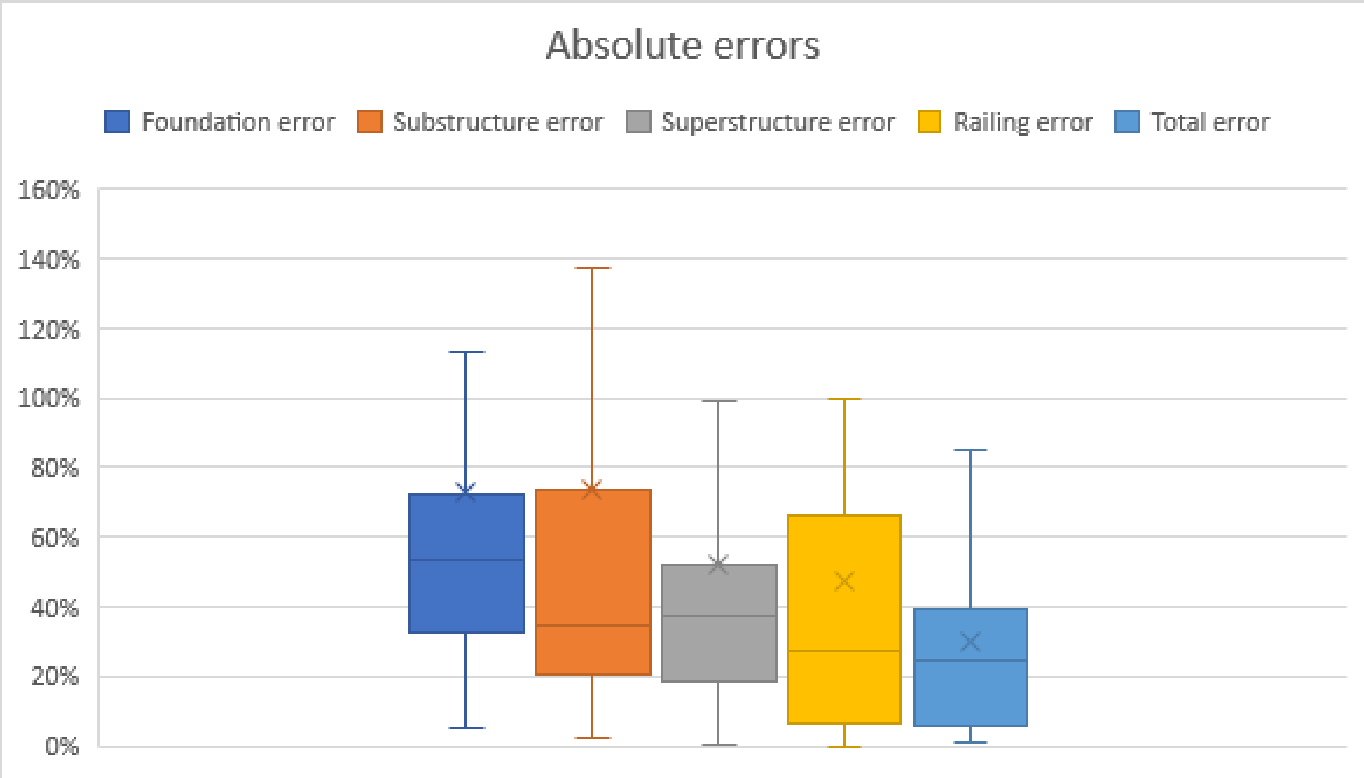

Figure 26: Absolute errors FALCON 5-Fold cross-validation

Table 17: Values Boxplots FALCON 5-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum	418%	416%			
Value above maximum	250%	403%		297%	120%
Value above maximum	199%	362%	497%	203%	91%
Maximum error	113%	138%	99%	100%	85%
Upper Quartile	72%	74%	52%	66%	40%
Average error 'X'	73%	74%	52%	47%	30%
Median	53%	35%	37%	27%	25%
Lower Quartile	33%	21%	19%	6%	6%
Minimum error	5%	2%	1%	0%	1%

Table 18: Absolute errors FALCON 5-fold cross-validation

	Absolute error foundation in %	Absolute error substructure in %	Absolute error superstructure in %	Absolute error railing in %	Absolute total error in %
Project 2	71%	35%	18%	203%	42%
Project 3	73%	99%	96%	0%	91%
Project 4	13%	2%	29%	4%	9%
Project 7	250%	16%	61%	61%	55%
Project 8	113%	60%	41%	6%	10%
Project 10	60%	35%	99%	34%	16%
Project 11	107%	24%	38%	100%	1%
Project 13	46%	5%	1%	94%	23%
Project 14	7%	2%	18%	23%	2%
Project 15	43%	47%	26%	33%	20%
Project 16	53%	59%	53%	48%	26%
Project 17	61%	31%	61%	17%	46%
Project 18	38%	48%	38%	21%	33%
Project 19	107%	26%	497%	7%	85%
Project 20	67%	31%	7%	297%	2%
Project 21	29%	78%	29%	20%	38%
Project 22	199%	20%	49%	79%	27%
Project 23	62%	403%	4%	21%	81%
Project 24	32%	4%	1%	49%	2%
Project 25	27%	54%	66%	86%	30%
Project 26	52%	85%	83%	64%	40%
Project 27	41%	362%	37%	0%	120%
Project 28	9%	138%	22%	67%	28%
Project 29	54%	416%	31%	0%	2%
Project 30	43%	26%	50%	32%	23%
Project 32	55%	47%	20%	1%	5%
Project 33	5%	19%	12%	0%	4%
Project 34	36%	32%	25%	1%	30%
Project 36	58%	4%	45%	34%	18%
Project 37	77%	30%	47%	9%	2%
Project 38	418%	80%	51%	16%	27%
Project 39	28%	35%	7%	84%	18%

5. Findings

Based on the presented results above, table 19 is formed to show the average errors per sample size. This table is shown below. It must be concluded that the total error is the lowest for N-fold cross-validation. In this case, each test project can be compared with the highest number of reference projects.

The error for the railing is another interesting topic. The error is smaller in the case of 5-fold cross-validation. An explanation for this may be that the fit with a project that at first sight matches less in terms of properties does match well in terms of price of the project.

Table 19: Average absolute errors different types of cross-validation

	Foundation	Substructure	Superstructure	Railing	Total
N-fold	61%	49%	54%	58%	24%
10-fold	64%	70%	55%	52%	29%
5-fold	73%	74%	52%	47%	30%

Appendix IV

Extensive explanation results multiple linear regression

This appendix shows all results and predictions for the multiple linear regression method. Per type of cross-validation, first, the results of the hyperparameter analysis are presented, followed by the real results. In the end, it also shows if the sample size has some influence on the results of this method.

Content

1. Hyperparameter analysis n-fold cross-validation
As mentioned in paragraph 6.1 a hyperparameter analysis is executed. Table 20 shows all test results. The final result is the most frequently occurring outcome. This result is used for the real model results.

Table 20: Results hyperparameter analysis multiple linear regression n-fold cross-validation

	Foundation		Substructure		Superstructure		Railing	
Fold	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize
1	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
2	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
3	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
4	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
5	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
6	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
7	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
8	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
9	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
10	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
11	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
12	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
13	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
14	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
15	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
16	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
17	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
18	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
19	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
20	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
21	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
22	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
23	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
24	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
25	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
26	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
27	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
28	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
29	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
30	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
31	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
32	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
33	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE
34	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
35	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
36	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
37	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
38	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
39	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE
Result	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE

2. Results n-fold cross-validation

Using the multiple linear regression module in Python, results are generated for each project. Table 21 shows these results. Before the absolute errors were calculated, negative values were replaced for zero. Six projects are excluded from the list, they are marked as an outlier. Project 5,6,9, 22, 26, and 27 are these projects. A cause for a misfit can be explained for projects 5,6,9 as mentioned in Appendix III.
The other projects are marked as extreme outliers and therefore removed.

Table 21: Absolute errors n-fold multiple regression analysis

	Absolute Error Foundation in %	Absolute error substructure in %	Absolute error superstructure in %	Absolute error railing in %	Absolute total error in %
Project 1	218%	100%	100%	44%	28%
Project 2	49%	17%	100%	22%	46%
Project 3	85%	77%	93%	0%	82%
Project 4	71%	38%	147%	71%	110%
Project 7	222%	88%	130%	31%	48%
Project 8	120%	60%	35%	161%	23%
Project 10	96%	55%	100%	161%	9%
Project 11	60%	66%	9%	89%	41%
Project 12	89%	100%	100%	55%	40%
Project 13	57%	49%	30%	359%	24%
Project 14	100%	254%	100%	15%	27%
Project 15	102%	113%	224%	53%	150%
Project 16	22%	34%	52%	34%	10%
Project 17	132%	87%	249%	2%	130%
Project 18	56%	11%	106%	27%	18%
Project 19	38%	84%	470%	19%	143%
Project 20	81%	87%	100%	79%	76%
Project 21	58%	89%	100%	11%	80%
Project 23	100%	267%	100%	16%	2%
Project 24	63%	67%	10%	164%	38%
Project 25	55%	683%	26%	33%	103%
Project 28	87%	106%	100%	35%	6%
Project 29	108%	629%	86%	37%	83%
Project 30	10%	60%	100%	88%	73%
Project 31	100%	4%	100%	0%	76%
Project 32	61%	89%	58%	9%	24%
Project 33	79%	100%	130%	38%	14%
Project 34	61%	100%	100%	4%	17%
Project 35	100%	1220%	100%	100%	76%
Project 36	100%	49%	166%	105%	24%
Project 37	100%	73%	495%	9%	176%
Project 38	1970%	100%	89%	53%	47%
Project 39	69%	100%	83%	19%	44%

3. Hyperparameter analysis 10-fold cross-validation

For the 10-fold cross-validation variant, a hyperparameter analysis is also executed. The results are shown in table 22.

Table 22: Results hyperparameter analysis multiple linear regression 10-fold cross-validation

	Foundation		Substructure		Superstructure		Railing	
Fold	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize
1	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
2	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
3	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
4	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
5	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
6	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
7	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
8	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
9	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE
10	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
Result	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE

4. Results 10-fold cross-validation

Using the multiple linear regression module in Python, results are generated for each project. Figure 27, table 23, and table 24 show the results. Before the absolute errors were calculated, negative values were replaced for zero. Values above a maximum of a boxplot are not visualized, these are shown in the corresponding table.

Six projects are excluded from the list, they are marked as an outlier. Project 5,6,9, 19, 22, and 26 are these projects. A cause for a misfit can be explained for projects 5,6,9 as mentioned in Appendix III.
The other projects are marked as extreme outliers and therefore removed.

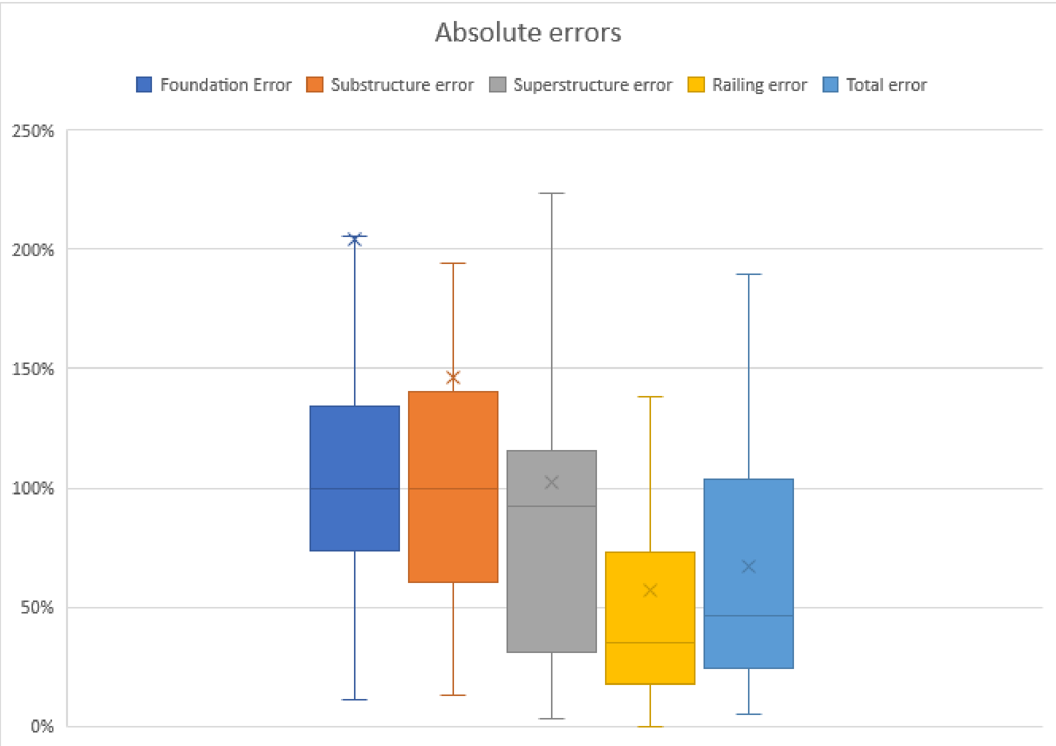

Figure 27: Absolute errors multiple linear regression 10-fold cross-validation

Table 23: Values boxplots multiple linear regression 10-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum		836%			
Value above maximum	2308%	546%			
Value above maximum	752%	400%	496%		
Value above maximum	622%	305%	318%	348%	
Value above maximum	292%	268%	285%	160%	
Maximum error	206%	194%	223%	139%	190%
Upper Quartile	134%	140%	115%	73%	104%
Average error 'X'	204%	146%	102%	57%	67%
Median	100%	100%	92%	35%	46%
Lower Quartile	74%	60%	31%	18%	25%
Minimum error	11%	13%	3%	0%	5%

Table 24: Absolute errors 10-fold multiple regression analysis

	Absolute Error Foundation in %	Absolute error substructure in %	Absolute error superstructure in %	Absolute error railing in %	Absolute total error in %
Project 1	292%	100%	100%	44%	14%
Project 2	95%	108%	25%	28%	22%
Project 3	137%	59%	92%	0%	76%
Project 4	81%	32%	165%	70%	117%
Project 7	206%	71%	107%	43%	84%
Project 8	107%	79%	34%	160%	10%
Project 10	77%	305%	44%	139%	77%
Project 11	71%	69%	10%	86%	46%
Project 12	191%	100%	5%	59%	21%
Project 13	33%	61%	28%	348%	27%
Project 14	100%	268%	100%	19%	32%
Project 15	124%	154%	285%	58%	190%
Project 16	11%	32%	4%	34%	12%
Project 17	132%	127%	318%	12%	167%
Project 18	56%	17%	91%	26%	11%
Project 20	98%	35%	9%	59%	46%
Project 21	60%	85%	45%	17%	66%
Project 23	100%	327%	100%	6%	13%
Project 24	47%	76%	3%	164%	43%
Project 25	44%	546%	10%	41%	116%
Project 27	43%	400%	112%	0%	189%
Project 28	97%	194%	100%	28%	28%
Project 29	130%	836%	119%	35%	111%
Project 30	622%	100%	41%	76%	55%
Project 31	100%	13%	100%	0%	72%
Project 32	98%	100%	97%	28%	39%
Project 33	90%	100%	173%	45%	28%
Project 34	100%	80%	53%	6%	33%
Project 35	752%	24%	42%	100%	160%
Project 36	100%	82%	223%	107%	43%
Project 37	100%	100%	496%	2%	166%
Project 38	2308%	100%	187%	21%	96%
Project 39	146%	49%	55%	22%	5%

5. Hyperparameter analysis 5-fold cross-validation

For the last variant, a hyperparameter analysis is also executed. The results are shown in table 25.

Table 25: Results hyperparameter analysis multiple linear regression 5-fold cross-validation

	Foundation		Substructure		Superstructure		Railing	
	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize	Fit intercept	Normalize
1	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
2	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	TRUE
3	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
4	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
5	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE
Result	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE

6. Results 5-fold cross-validation

Using the multiple linear regression module in Python, results are generated for each project. Figure 28, table 26, and table 27 show the results. Before the absolute errors were calculated, negative values were replaced for zero. Values above a maximum of a boxplot are not visualized, these are shown in the corresponding table.

Six projects are excluded from the list, they are marked as an outlier. Project 5, 9, 11, 19, 22, 35, and 39 are these projects. A cause for a misfit can be explained for projects 5, 9, and 35 as mentioned in Appendix III.
The other projects are marked as extreme outliers and therefore removed.

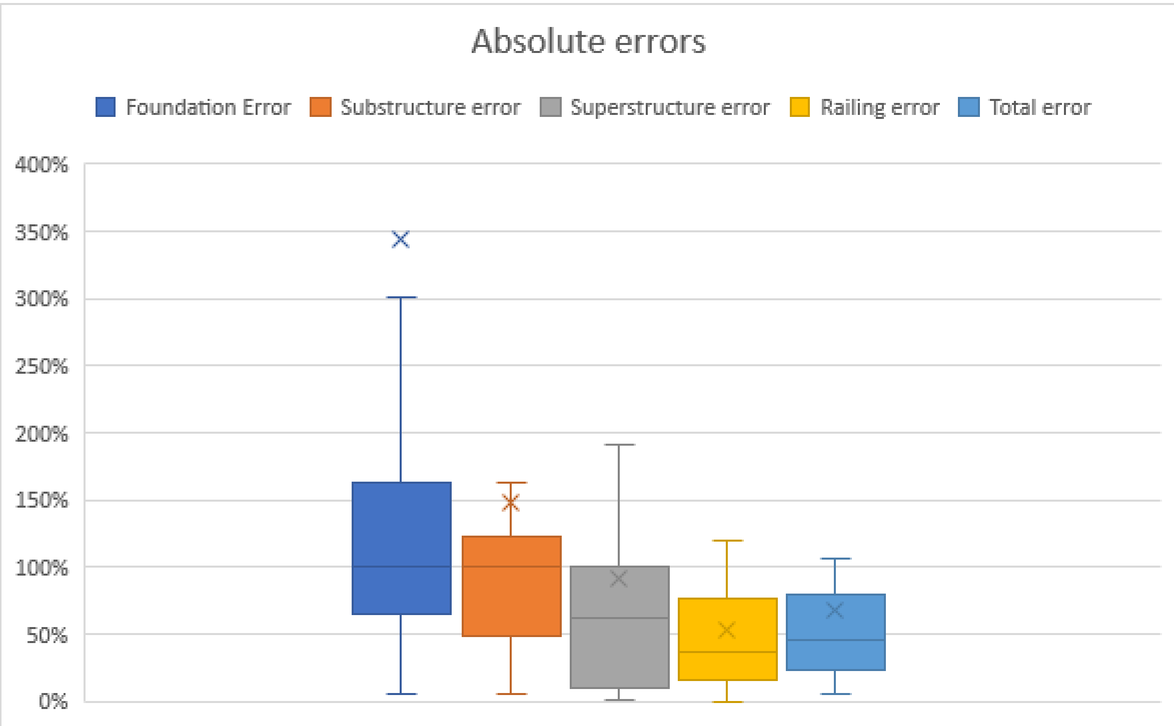

Figure 28: Absolute errors multiple linear regression 5-fold cross-validation

Table 26: Values boxplots multiple linear regression 5-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum		882%			
Value above maximum	5301%	559%	519%		
Value above maximum	1312%	443%	320%		222%
Value above maximum	953%	415%	291%	235%	210%
Value above maximum	680%	316%	275%	177%	167%
Maximum error	302%	163%	191%	119%	107%
Upper Quartile	162%	122%	100%	77%	79%
Average error 'X'	344%	148%	92%	53%	68%
Median	100%	100%	61%	37%	46%
Lower Quartile	65%	49%	9%	15%	23%
Minimum error	6%	5%	1%	0%	6%

Table 27: Absolute errors 5-fold multiple regression analysis

	Absolute Error Foundation in %	Absolute error substructure in %	Absolute error superstructure in %	Absolute error railing in %	Absolute total error in %
Project 1	191%	100%	50%	45%	20%
Project 2	100%	443%	5%	16%	68%
Project 3	172%	19%	99%	0%	73%
Project 4	74%	43%	8%	75%	47%
Project 6	1312%	882%	74%	100%	210%
Project 7	259%	100%	82%	173%	44%
Project 8	103%	80%	60%	88%	15%
Project 10	130%	316%	1%	78%	107%
Project 12	302%	100%	8%	43%	35%
Project 13	6%	68%	14%	177%	32%
Project 14	100%	122%	100%	14%	9%
Project 15	90%	147%	291%	62%	189%
Project 16	17%	38%	19%	40%	21%
Project 17	119%	122%	320%	15%	167%
Project 18	46%	20%	7%	35%	19%
Project 20	88%	5%	4%	69%	28%
Project 21	62%	81%	4%	19%	53%
Project 23	100%	99%	100%	1%	81%
Project 24	34%	42%	31%	235%	6%
Project 25	12%	415%	73%	18%	43%
Project 26	953%	100%	100%	19%	48%
Project 27	12%	559%	32%	0%	175%
Project 28	133%	163%	100%	38%	30%
Project 29	131%	100%	28%	28%	10%
Project 30	680%	100%	62%	78%	54%
Project 31	92%	17%	100%	0%	66%
Project 32	94%	100%	109%	24%	48%
Project 33	87%	100%	191%	41%	37%
Project 34	31%	17%	7%	2%	34%
Project 26	85%	76%	60%	119%	17%
Project 37	100%	100%	519%	11%	177%
Project 38	5301%	75%	275%	32%	222%

7. Findings

Based on the presented results above, table 28 is formed to show the average errors per sample size. This table is shown below. It must be concluded that the total error is the lowest for n-fold cross-validation. In this case, the multiple linear regression calculation is made based on the highest number of reference projects. The average deviation per category is quite large for each category.

Table 28: Average absolute errors different types of cross-validation

	Foundation	Substructure	Superstructure	Railing	Total
N-fold	143%	153%	121%	59%	57%
10-fold	204%	146%	102%	57%	67%
5-fold	344%	148%	92%	53%	68%

Appendix V

Extensive explanation results decision trees

In this appendix, the results for the decision tree results are given. All results are given for the hyperparameter analysis and the results in case the amount of folds is equal to 5, 10, and 39. As result, it can be concluded which parameters and fold size give the best solution.

Content

1. Hyperparameter analysis n-fold cross-validation
2. Results n-fold cross-validation
3. Hyperparameter analysis 10-fold cross-validation
4. Results 10-fold cross-validation
5. Hyperparameter analysis 5-fold cross-validation
6. Results 5-fold cross-validation
7. Findings

1. Hyperparameter analysis n-fold cross-validation

Before the results with the use of the decision tree are generated using n-fold cross-validation, a hyperparameter optimization was done. Table 29 shows the results of the hyperparameter analysis.

Table 29: Results 39Fold hyperparameter analysis decision tree

Fold	Max depth	Min samples leaf	Min samples split
1	6	4	9
2	4	1	2
3	4	1	2
4	4	1	2
5	5	2	2
6	4	1	4
7	5	3	7
8	4	1	4
9	4	1	3
10	3	1	3
11	6	4	2
12	4	1	3
13	5	4	9
14	4	1	2
15	4	1	2
16	4	1	2
17	4	1	2
18	4	1	2
19	4	1	2
20	4	1	2
21	4	1	2
22	4	1	2
23	4	1	2
24	4	1	2
25	4	1	2
26	4	1	2
27	5	1	2
28	4	1	2
29	4	1	2
30	4	1	2
31	4	1	3
32	4	1	2
33	5	4	9
34	4	1	2
35	6	4	2
36	3	1	5
37	3	1	5
38	4	1	2
39	4	1	2
Result	4	1	2

2. Results n-fold cross-validation

After finishing the hyperparameter optimization, results are generated for the total price of a project and presented in table 30. In the testing phase, it became clear that the best results with this method are generated in case the total price is predicted. Therefore, only the total price, the most important price, is predicted.

Table 30: Absolute errors decision tree with n-fold cross-validation

Project number	Abs. Error
Project 1	21%
Project 2	17%
Project 3	92%
Project 4	18%
Project 5	91%
Project 6	88%
Project 7	8%
Project 8	48%
Project 10	61%
Project 11	34%
Project 12	13%
Project 13	15%
Project 14	24%
Project 17	85%
Project 19	7%
Project 20	37%
Project 21	77%
Project 22	7%
Project 23	25%
Project 24	8%
Project 26	23%
Project 26	103%
Project 27	21%
Project 28	20%
Project 28	58%
Project 30	14%
Project 31	57%
Project 32	28%
Project 33	34%
Project 34	17%
Project 35	39%
Project 36	6%
Project 37	17%
Project 38	23%
Project 39	6%
Average	36%

3. Hyperparameter analysis 10-fold cross-validation

The same method for hyperparameter analysis is applied to the 5-fold case. Results of this hyperparameter analysis are shown in table 31. The end result is equal to the result of the hyperparameter analysis of the 39-fold case.

Table 31: Results 5Fold hyperparameter analysis decision tree

10Fold			
Fold	Max depth	Min samples leaf	Min samples split
1	5	1	2
2	7	3	7
3	4	1	2
4	8	1	3
5	3	1	5
6	4	1	2
7	4	1	2
8	4	1	5
9	5	4	2
10	4	1	2
Result	4	1	2

4. Results 10-fold cross-validation

Using the optimal parameters, the results are generated. These results are corrected for outliers and the results are shown in figure 29. Results from projects 9, 15,19,25, and 39 are omitted. Project 9 is a wooden bridge. The other projects are extreme outliers and therefore removed. Table 32 shows the results per project.

The lowest value is 2%, the lower quartile value is 16%, the median 23%, the average value is 35% (x), the upper quartile value is 46% and the highest value is 85%. Three values are above the maximum value of the boxplot: 93%, 95%, and 103%.

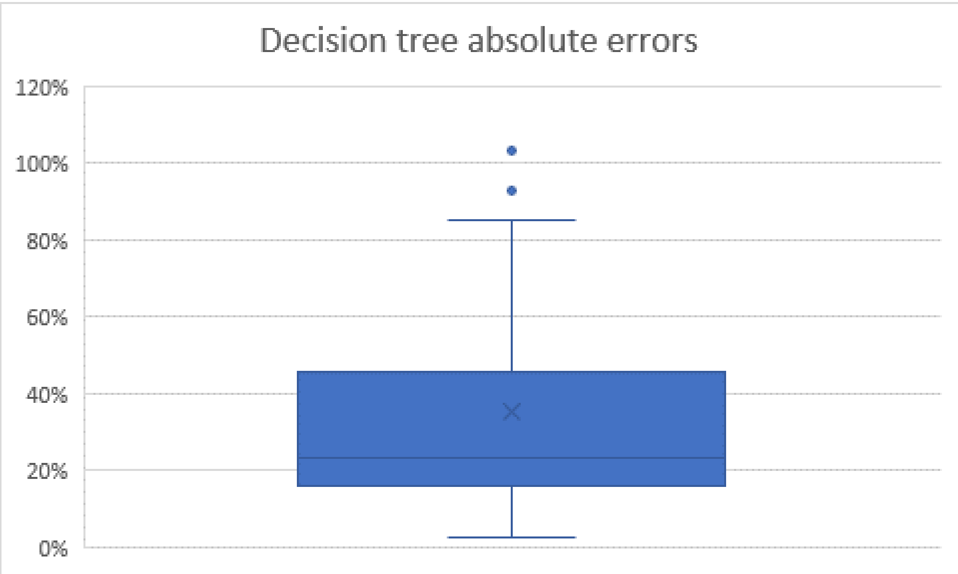

Figure 29: Results decision tree 10-Fold cross-validation

Table 32: Absolute errors decision tree with 10-fold cross-validation

Project	Absolute error
Project 1	21%
Project 2	16%
Project 3	93%
Project 4	18%
Project 5	95%
Project 6	41%
Project 7	16%
Project 8	48%
Project 10	21%
Project 11	33%
Project 12	13%
Project 13	32%
Project 14	24%
Project 16	103%
Project 17	85%
Project 18	63%
Project 20	45%
Project 21	80%
Project 22	7%
Project 23	25%
Project 24	12%
Project 26	18%
Project 27	23%
Project 28	11%
Project 29	18%
Project 30	14%
Project 31	57%
Project 32	34%
Project 33	40%
Project 34	2%
Project 35	39%
Project 36	6%
Project 37	17%
Project 38	23%

5. Hyperparameter analysis 5-fold cross-validation

For the 5-fold case, the hyperparameter analysis is executed. As shown in table 33, the maximum depth is not clear. The value for the maximum depth is set equal to the 10-fold and n-fold settings. The maximum depth is thus 4.

Table 33: Results 5Fold hyperparameter analysis decision tree

5Fold			
Fold	Max depth	Min samples leaf	Min samples split
1	3	3	2
2	8	1	3
3	2	1	5
4	5	1	7
5	4	3	7
Result	?	1	7

6. Results 5-fold cross-validation

Using the optimal parameters as shown in the former paragraph, the results are calculated. Table 34 shows the numerical values. In figure 30, the results without outliers are visualized. The results of projects 9, 15, 16, and 23 are omitted. Project 9 is as already mentioned the only wooden bridge. Project 15,16, and 23 are extreme outliers in this case.

The lowest value is 2%, the lower quartile value is 17%, the median 28%, the average value is 36% (x), the upper quartile value is 53% and the highest value is 97%.

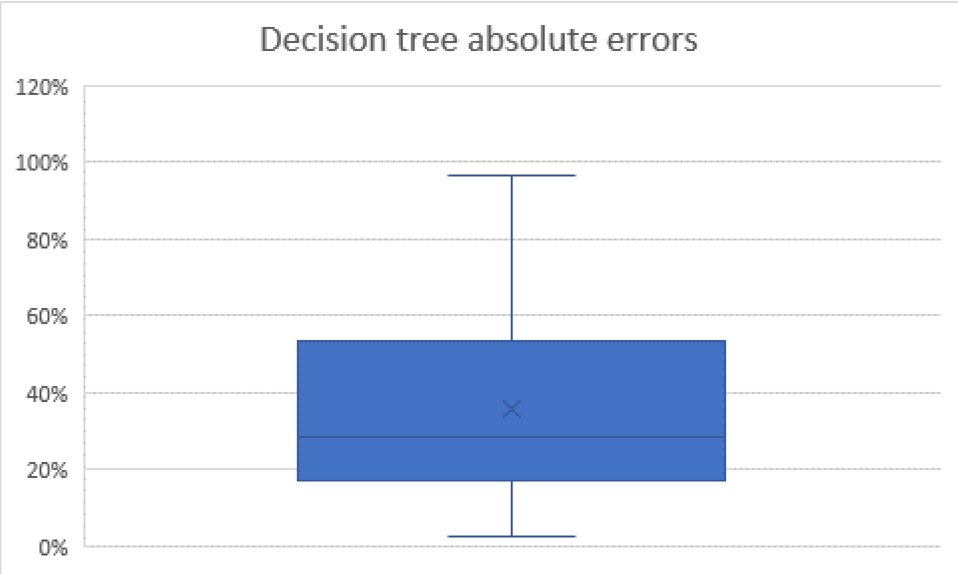

Figure 30: Results decision tree 5-fold cross-validation

Table 34: Absolute errors decision tree with 5-fold cross-validation

Project	Absolute error
Project 1	69%
Project 2	18%
Project 3	97%
Project 4	26%
Project 5	22%
Project 6	66%
Project 7	53%
Project 8	48%
Project 10	21%
Project 11	33%
Project 12	10%
Project 13	32%
Project 14	21%
Project 17	72%
Project 18	70%
Project 19	35%
Project 20	45%
Project 21	80%
Project 22	7%
Project 24	60%
Project 25	18%
Project 26	28%
Project 27	14%
Project 28	14%
Project 29	19%
Project 30	14%
Project 31	56%
Project 32	39%
Project 33	46%
Project 34	2%
Project 35	39%
Project 36	16%
Project 37	17%
Project 38	23%
Project 39	13%

7. Findings

In the case of n-fold cross-validation, the average error is 36%. In the case of 10-fold cross-validation, the average error is 35%. In the case of 5-fold cross-validation, the average error is 36%. Based on these results, the error does not vary a lot when the sample size varies.

Appendix VI

Extensive explanation results KNN

This appendix shows all results and predictions for the multiple linear regression method. Per type of cross-validation, first, the results of the hyperparameter analysis were presented, followed by the real results. In the end, it also shows if the sample size had some influence on the results of this method.

Content

1. Hyperparameter analysis n-fold cross-validation

As mentioned in paragraph 6.3 a hyperparameter analysis is executed for each category. The results are shown in table 35. The result is the most frequently occurring outcome. This result is used for the real model results.
In this table: N = number of neighbors, CS = Chebyshev, MH = Manhattan and EC = Euclidean.

Table 35: Results n-fold hyperparameter analysis KNN

	Foundation				Substructure				Superstructure				Railing			
39 Fold	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	n neigh-bors	Algo-rithm	leaf size	metric	N
1	ball	18	CS	7	ball	9	CS	1	brute	1	CS	6	ball	1	EC	5
2	ball	18	CS	4	ball	9	CS	1	ball	17	CS	2	ball	1	EC	5
3	brute	1	CS	5	ball	9	CS	1	ball	9	CS	2	ball	1	MH	6
4	kd	17	CS	7	ball	9	CS	1	brute	1	CS	6	ball	1	MH	7
5	ball	18	CS	5	ball	9	CS	1	ball	9	CS	2	ball	1	EC	5
6	ball	17	CS	7	ball	9	CS	1	ball	9	CS	2	ball	1	MH	8
7	ball	18	CS	4	ball	9	CS	1	ball	17	CS	2	ball	1	EC	5
8	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	EC	5
9	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	MH	6
10	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	EC	5
11	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	MH	7
12	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	MH	8
13	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	EC	5
14	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	3	CS	3
15	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	3	CS	2
16	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	MH	6
17	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	CS	2
18	ball	18	CS	4	ball	17	CS	2	ball	9	CS	2	ball	1	CS	2
19	ball	18	CS	4	ball	17	CS	2	kd	9	CS	2	ball	1	CS	2
20	ball	18	CS	4	ball	18	CS	2	ball	9	CS	2	ball	1	CS	2

21	ball	18	CS	8	ball	9	CS	3	ball	9	CS	2	ball	1	CS	2
22	ball	9	CS	7	kd	9	CS	1	ball	9	CS	2	ball	1	CS	2
23	ball	18	CS	5	kd	9	CS	9	ball	9	CS	2	ball	1	CS	2
24	ball	18	CS	5	kd	9	CS	9	ball	9	CS	2	ball	1	CS	2
25	brute	1	CS	5	kd	9	CS	9	ball	9	CS	2	ball	1	CS	2
26	ball	18	CS	4	kd	9	CS	9	ball	17	CS	2	ball	1	CS	2
27	ball	18	CS	4	kd	9	CS	9	ball	9	CS	2	ball	1	CS	2
28	brute	1	CS	5	kd	9	CS	9	ball	18	CS	1	ball	1	CS	2
29	brute	1	CS	5	kd	9	CS	9	ball	17	CS	2	ball	1	CS	2
30	ball	18	CS	5	ball	18	CS	2	ball	9	CS	2	ball	1	CS	2
31	ball	18	CS	4	brute	1	CS	5	ball	17	CS	2	ball	1	CS	2
32	ball	18	CS	4	brute	1	CS	5	kd	9	CS	2	ball	1	CS	2
33	brute	1	CS	7	ball	18	CS	2	ball	17	CS	2	ball	1	CS	2
34	kd	17	CS	4	kd	9	CS	1	kd	17	CS	2	ball	1	CS	2
35	kd	17	CS	4	kd	9	CS	1	kd	17	CS	2	ball	3	CS	3
36	kd	17	CS	4	kd	9	CS	1	kd	17	CS	2	ball	1	CS	3
37	kd	17	CS	4	kd	17	CS	2	kd	17	CS	2	ball	1	CS	2
38	kd	17	CS	4	ball	9	CS	3	kd	17	CS	2	ball	1	CS	2
39	ball	9	CS	6	brute	1	CS	8	brute	1	CS	7	ball	2	CS	3
Re-sult	ball	18	CS	4	ball	9	CS	1	ball	9	CS	2	ball	1	CS	2

2. Results n-fold cross-validation

Using the multiple linear regression method in Python, results are generated for each project. These results are shown in table 36. Before the absolute errors were calculated, negative values were replaced for zero. Seven projects are excluded from the list, they are marked as an outlier. Projects 5, 9, 12, 21, 22, 27, and 37 are these projects. A cause for a misfit can be explained for projects 5 and 9 as mentioned in Appendix III.

The other projects are marked as extreme outliers and therefore removed.

Table 36: Absolute errors KNN with n-fold cross-validation

	Foundation error	Substructure error	Superstructure error	Railing error	Total error
Project 1	68%	5%	62%	24%	60%
Project 2	59%	45%	15%	16%	49%
Project 3	60%	92%	97%	0%	94%
Project 4	20%	2%	25%	52%	8%
Project 6	17%	67%	73%	156%	32%
Project 7	106%	16%	65%	128%	29%
Project 8	7%	55%	23%	78%	25%
Project 10	19%	82%	119%	169%	69%
Project 11	56%	37%	17%	52%	3%
Project 12	98%	55%	17%	91%	56%
Project 13	0%	5%	28%	306%	2%
Project 15	83%	47%	8%	12%	28%
Project 16	46%	21%	38%	19%	25%
Project 17	42%	31%	51%	13%	39%
Project 18	22%	27%	3%	8%	9%
Project 19	363%	46%	278%	140%	87%
Project 20	57%	17%	3%	86%	16%
Project 23	26%	31%	79%	21%	47%
Project 24	39%	4%	44%	41%	23%
Project 25	33%	20%	59%	62%	38%
Project 26	125%	37%	64%	4%	6%
Project 28	73%	50%	15%	47%	27%
Project 29	43%	672%	20%	24%	11%
Project 30	70%	26%	15%	87%	1%
Project 31	82%	72%	10977%	0%	97%
Project 32	81%	23%	24%	13%	5%
Project 33	1%	19%	26%	25%	2%
Project 34	7%	32%	7%	20%	29%
Project 35	303%	395%	8%	38%	109%
Project 36	52%	30%	30%	5%	18%
Project 38	2481%	147%	46%	40%	38%
Project 39	53%	85%	50%	41%	6%

3. Hyperparameter analysis 10-fold cross-validation

For the 10-fold cross-validation variant, a hyperparameter analysis is also executed. The results are shown in table 37. In this table: N = number of neighbors, CS = Chebyshev, MH = Manhattan and EC = Euclidean.

Table 37: Results 10-fold hyperparameter analysis KNN

	Foundation				Substructure				Superstructure				Railing			
10 Fold	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N
1	brute	1	CS	4	ball	8	CS	1	ball	8	CS	4	ball	2	CS	5
2	ball	16	CS	5	ball	8	CS	1	ball	8	CS	2	ball	2	CS	2
3	ball	16	CS	4	kd	8	CS	1	ball	8	CS	2	ball	2	CS	2
4	ball	16	CS	9	ball	8	CS	1	ball	8	CS	2	ball	1	MH	7
5	ball	16	CS	5	ball	8	CS	1	ball	8	CS	2	ball	1	EC	6
6	ball	16	CS	8	ball	16	CS	8	kd	8	CS	2	ball	1	MH	9
7	ball	16	CS	5	ball	8	CS	1	ball	16	CS	1	ball	1	MH	4
8	ball	16	CS	5	ball	8	CS	1	brute	1	CS	4	ball	1	MH	7
9	ball	16	CS	4	kd	8	CS	1	brute	1	CS	7	ball	1	CS	2
10	brute	1	CS	4	brute	1	CS	7	kd	9	CS	3	ball	3	CS	6
Re-sult	ball	16	CS	5	ball	8	CS	1	ball	8	CS	2	ball	1	CS	2

4. Results 10-fold cross-validation

Using the KNN module in Python, results are generated for each project. Figure 31, table 38, and table 39 show the results. Before the absolute errors were calculated, negative values were replaced for zero. Values above a maximum of a boxplot are not visualized, these are shown in the corresponding table.

Five projects are excluded from the list, they are marked as an outlier. Projects 3,5,9,27, and 35 are these projects. A cause for a misfit can be explained for project 5,9 and 35 as mentioned in Appendix III.
The other projects are marked as extreme outliers and therefore removed.

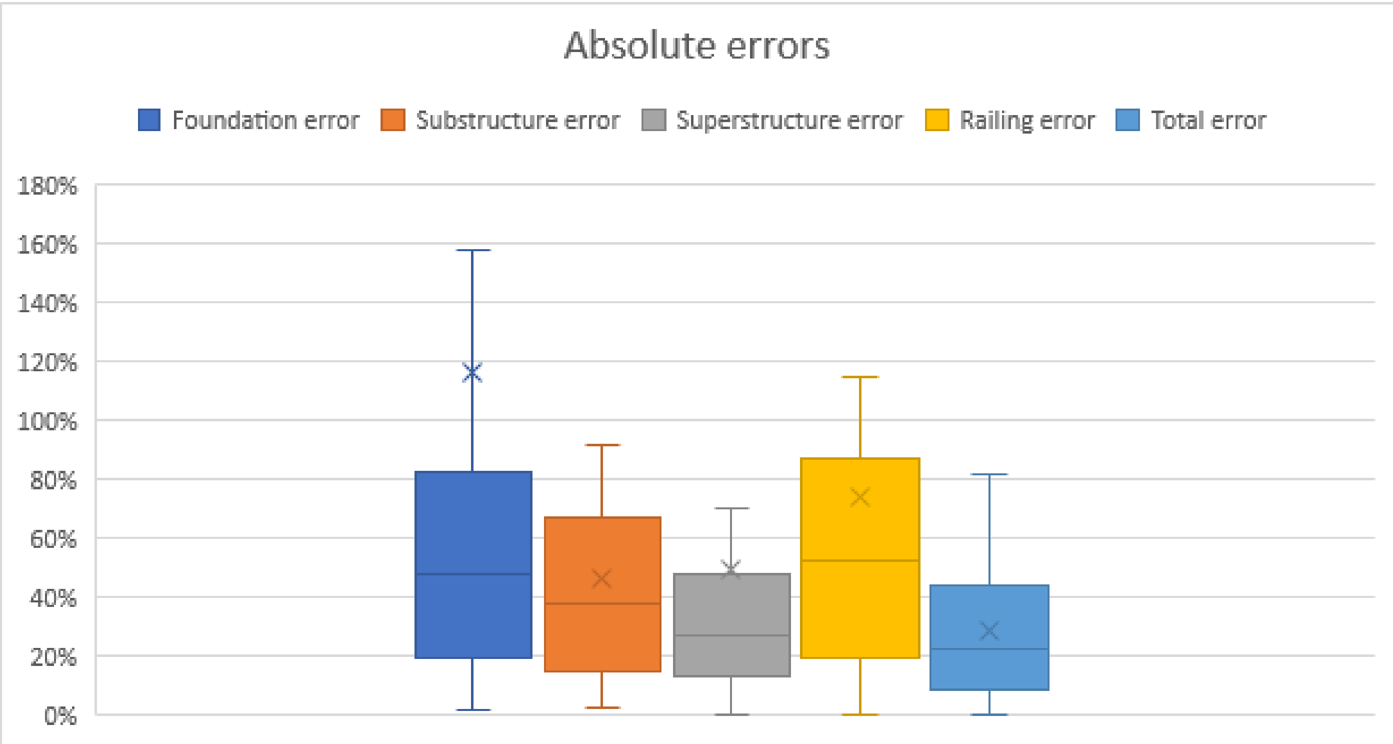

Figure 31: Absolute errors KNN 10-fold cross-validation

Table 38: Values boxplots KNN 10-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum	1955%		401%	652%	
Value above maximum	303%	258%	327%	241%	
Maximum error	158%	92%	70%	115%	82%
Upper Quartile	83%	67%	48%	87%	44%
Average error 'X'	116%	46%	50%	74%	29%
Median	48%	38%	27%	52%	23%
Lower Quartile	20%	15%	13%	19%	8%
Minimum error	2%	2%	0%	0%	0%

Table 39: Absolute errors 10-fold KNN

	Foundation error	Substructure error	Superstructure error	Railing error	Total error
Project 1	4%	61%	45%	24%	50%
Project 2	62%	69%	15%	16%	57%
Project 4	40%	2%	25%	52%	10%
Project 6	96%	83%	70%	81%	29%
Project 7	82%	16%	65%	79%	21%
Project 8	20%	75%	23%	107%	37%
Project 10	3%	2%	58%	98%	21%
Project 11	2%	65%	17%	87%	31%
Project 12	82%	13%	26%	91%	43%
Project 13	18%	5%	23%	652%	8%
Project 14	154%	2%	13%	12%	20%
Project 15	141%	47%	35%	12%	45%
Project 16	33%	21%	38%	19%	24%
Project 17	92%	31%	51%	13%	44%
Project 18	6%	27%	3%	8%	8%
Project 19	303%	51%	327%	91%	78%
Project 20	63%	54%	3%	86%	29%
Project 21	61%	92%	42%	42%	72%
Project 22	12%	20%	29%	53%	0%
Project 23	36%	91%	61%	21%	55%
Project 24	48%	4%	25%	241%	8%
Project 25	59%	56%	66%	55%	55%
Project 26	37%	45%	14%	74%	15%
Project 28	13%	16%	15%	47%	3%
Project 29	47%	258%	9%	24%	4%
Project 30	47%	26%	46%	87%	19%
Project 31	85%	84%	401%	0%	82%
Project 32	158%	47%	35%	44%	5%
Project 33	58%	19%	10%	57%	13%
Project 34	12%	32%	7%	20%	31%
Project 36	50%	4%	41%	16%	6%
Project 37	34%	74%	47%	74%	36%
Project 38	1955%	67%	1%	115%	18%
Project 39	47%	10%	0%	26%	4%

5. Hyperparameter analysis 5-fold cross-validation
For the last variant, a hyperparameter analysis is also executed. The results are shown in table 40. In this table: N = number of neighbors, CS = Chebyshev, MH = Manhattan and EC = Euclidean.

Table 40: Results 5-fold hyperparameter analysis KNN

	Foundation				Substructure				Superstructure				Railing			
5 Fold	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N	Algo-rithm	leaf size	metric	N
1	ball	1	EC	4	ball	7	CS	1	ball	1	CS	1	ball	14	CS	1
2	ball	14	CS	9	ball	7	CS	1	ball	2	CS	6	ball	1	MH	6
3	ball	14	CS	5	kd	7	CS	2	ball	14	CS	1	ball	1	MH	6
4	ball	14	CS	6	brute	1	CS	8	brute	1	CS	6	ball	1	MH	6
5	ball	15	CS	4	ball	15	CS	2	kd	14	CS	2	ball	1	EC	8
Re-sult	ball	14	CS	4	ball	7	CS	1	ball	14	CS	6	ball	1	MH	6

7. Results 5-fold cross-validation
Using the KNN module in Python, results are generated for each project. Figure 32, table 41, and table 42 show the results. Before the absolute errors were calculated, negative values were replaced for zero. Values above a maximum of a boxplot are not visualized, these are shown in the corresponding table.

Five projects are excluded from the list, they are marked as an outlier. Projects 3,5,9,27, and 35 are these projects. A cause for a misfit can be explained for projects 5,9, and 35 as mentioned in Appendix III. The other projects are marked as extreme outliers and therefore removed.

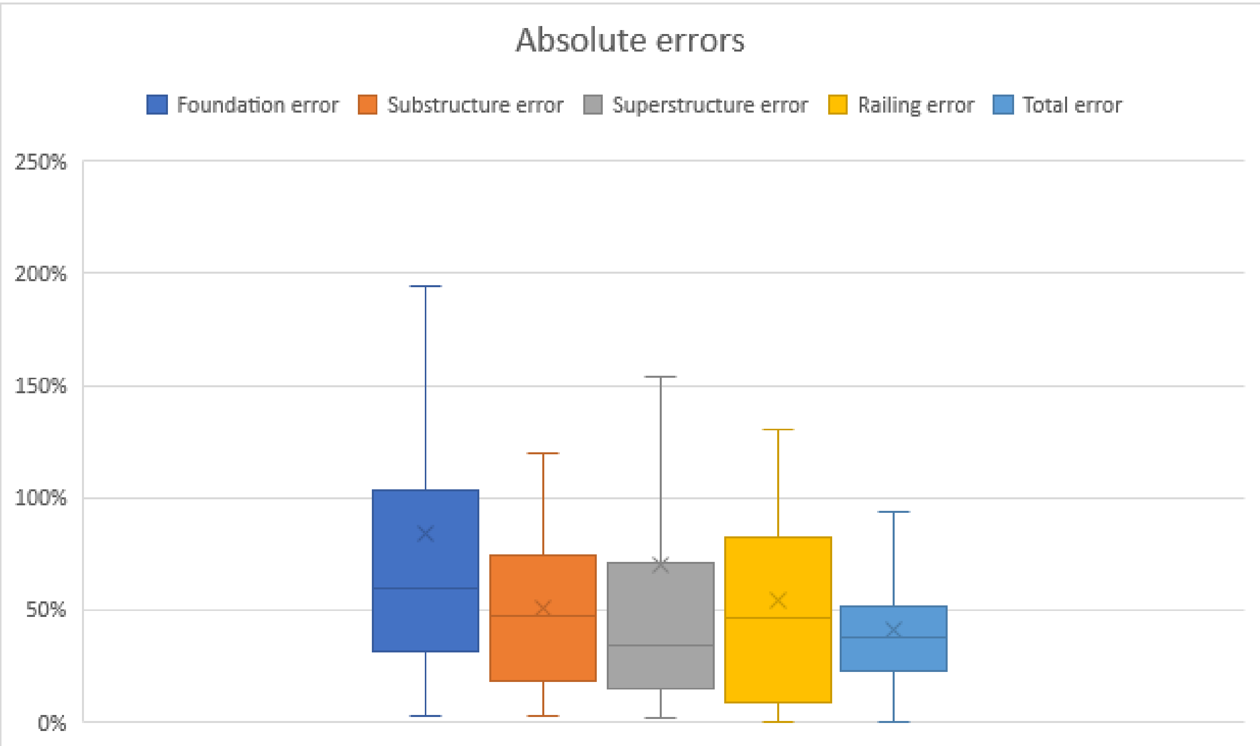

Figure 32: Absolute errors KNN 5-fold cross-validation

Table 41: Values boxplots KNN 5-fold cross-validation

	Foundation	Substructure	Superstructure	Railing	Total error
Value above maximum	426%		718%		110%
Value above maximum	322%	201%	315%	280%	108%
Maximum error	194%	120%	154%	130%	93%
Upper Quartile	103%	74%	70%	82%	51%
Average error 'X'	84%	50%	70%	55%	41%
Median	59%	47%	34%	46%	38%
Lower Quartile	31%	18%	14%	8%	22%
Minimum error	3%	2%	2%	0%	0%

Table 42: Absolute errors 5-fold KNN

	Foundation error	Substructure error	Superstructure error	Railing error	Total error
Project 1	73%	61%	39%	0%	59%
Project 2	59%	69%	2%	62%	50%
Project 3	60%	89%	97%	0%	93%
Project 4	20%	2%	15%	46%	25%
Project 6	46%	83%	73%	49%	46%
Project 7	112%	16%	95%	280%	46%
Project 8	7%	75%	16%	100%	39%
Project 9	63%	201%	58%	8%	37%
Project 10	3%	120%	116%	63%	65%
Project 11	56%	65%	12%	100%	19%
Project 12	94%	13%	27%	91%	46%
Project 13	157%	5%	25%	113%	19%
Project 14	194%	2%	61%	32%	37%
Project 15	130%	47%	68%	9%	52%
Project 16	35%	59%	37%	30%	38%
Project 17	84%	31%	88%	7%	50%
Project 18	15%	48%	17%	24%	35%
Project 20	57%	7%	5%	130%	6%
Project 21	37%	81%	34%	5%	56%
Project 22	30%	20%	2%	100%	14%
Project 23	3%	53%	315%	37%	108%
Project 24	39%	4%	21%	53%	12%
Project 25	33%	56%	56%	73%	43%
Project 26	125%	45%	2%	71%	9%
Project 28	59%	81%	28%	38%	36%
Project 30	70%	26%	14%	111%	2%
Project 31	84%	89%	718%	0%	75%
Project 32	322%	47%	2%	4%	40%
Project 33	158%	19%	37%	26%	29%
Project 34	7%	32%	6%	4%	32%
Project 35	426%	36%	44%	53%	110%
Project 36	74%	4%	154%	20%	0%
Project 37	23%	74%	24%	61%	36%

7. Findings

Based on the presented results above, table 43 is formed showing the average errors per sample size. This table is shown below. It has been concluded that the total error is the lowest for 10-fold cross-validation. As shown in Appendix III, the deviations from the predictions for the total price are in this case equal to the deviations from the predictions of FALCON.
The average error for the superstructure in case of n-fold cross-validation depends on one single project that has an outlier in the substructure case.

Table 43: Average absolute errors different types of cross-validation

	Foundation	Substructure	Superstructure	Railing	Total
N-fold	144%	72%	387%	57%	34%
10-fold	116%	46%	50%	74%	29%
5-fold	84%	50%	70%	55%	41%

Appendix VII

Questionnaire validation interviews

For the validation interviews, a set of questions is asked to the colleagues of Witteveen+Bos. These questions are shown below. These questions are in Dutch since all colleagues are speaking the Dutch language. Before these questions were asked to the respondents, a summary with information about the model was sent to the respondents. If there were questions about this information, extra information was given in that case before the interview started.

Question:

1. Als je op dit moment een brug moet ramen, in welke mate maak je dan gebruik van referentie projecten?
2. Als je gebruik maakt van referentieprojecten, hoe verkrijg je deze dan?
3. Zie jij mogelijkheden voor het gebruik van (AI) modellen, bijvoorbeeld FALCON, die kunnen helpen voor het maken van ramingen?
4. Als je mogelijkheden ziet voor het gebruik van (AI) modellen, bijvoorbeeld FALCON; in welke fase van een project zou je deze willen gebruiken?
5. Als jij het FALCON-model zou gebruiken voor een kostenschatting, welke gemiddelde nauwkeurigheid verwacht je dan en verschilt dit per fase in een project (SO/VO/DO)?
6. Zou je naar aanleiding van de informatie die je gekregen hebt over het FALCON-model in combinatie met de toelichtingen die gegeven zijn het FALCON-model gaan toepassen in de praktijk?
    - 6.1 Als het antwoord ja is, op welke manier?
    - 6.2 Als het antwoord nee is, waarom niet?
7. Zijn er nog verbeteringen mogelijk? Hierbij kun je denken aan het toevoegen van parameters, of andere informatie die volgens jou van belang is, of het uitsluiten van bepaalde categorieën?
8. In geval je het model op dit moment nog niet accuraat genoeg vindt, zou je gebruik maken van het model in geval de betrouwbaarheid van het model toeneemt?
    - 8.1 Zo ja, in hoeverre moet de betrouwbaarheid toenemen?
9. Het model vergelijkt zoals je weet projecten met elkaar, zou het van waarde zijn dat het model aangeeft welke projecten er zijn gebruikt voor het bepalen van de kostenschatting?
10. Heb je nog andere vragen of opmerkingen?