AN ARTIFICIAL NEURAL NETWORK APPROACH FOR COST ESTIMATION OF ENGINEERING SERVICES

ENHANCING COST ESTIMATING EFFICIENCY

UNIVERSITY OF TWENTE.

MASTER THESIS

Author

E. (Erik) Matel BSc
University of Twente
Department of Construction Management and Engineering

Date

Monday, May 27, 2019

Version

Final

BILFINGER TEBODIN

BILFINGER

“All models are wrong, but some are useful”

- George E.P. Box

Colophon

An artificial neural network approach for cost estimation of engineering services

Enhancing cost estimating efficiency

Master Thesis

Monday, May 27, 2019

Author

E. (Erik) Matel BSc
S1867482
Department of Construction Management and Engineering
University of Twente

Contact

Email: e.matel@student.utwente.nl
Tel: +31 (0)6 15 62 97 14

Company Supervisors

T. (Thijs) Evers MSc
Coordinator of Tender Management in Hengelo
Bilfinger Tebodin

W. (Willem) de Vries MSc
Coordinator of Tender Management in North West Europe
Bilfinger Tebodin

University Supervisors

dr. J.T. (Hans) Voordijk
Associate professor
Department of Construction Management and Engineering
University of Twente

dr.ir. F. (Farid) Vahdatikhaki
Assistant Professor
Department of Construction Management and Engineering
University of Twente

dr.ir. S. (Siavash) Hosseinyalamdary
Assistant Professor
Department of Earth Observation Science
University of Twente

PREFACE

This document contains the master's thesis "An artificial neural network approach for cost estimation of engineering services: enhancing cost estimation efficiency". This document is the formal document towards my graduation for the Master of Science study Construction Management and Engineering at the University of Twente. The report is intended to provide detailed information about the research that was conducted to establish the artificial neural network model. During the past seven months, I have developed a neural network that can be used to estimate the cost of engineering services. This research is carried out for Bilfinger Tebodin Hengelo under the supervision of Thijs Evers and Willem de Vries. Furthermore, this research is supervised by Farid Vahdatikhaki, Siavash Hosseinyalamdary and Hans Voordijk from the University of Twente.

I would like to use this preface to thank a number of people. First of all, I want to heartily thank my supervisors on behalf of the University for their constructive criticism and quick and prompt feedback during the research process. I have experienced the collaboration within the university committee as very decisive and purposeful. Moreover, I would also like to thank my supervisors of Bilfinger Tebodin for the daily supervision, good support, guidance, feedback and suggestion of new ideas. During this research project, I had the feeling that you always made time and always thought along well. Finally, I would like to thank all Bilfinger Tebodin colleagues for their good cooperation, support and working atmosphere during the past seven months.

Erik Matel

Hengelo, 2019

ABSTRACT

The expected pace for the completion of tenders which engineering consultancy firms need to perform is increasing rapidly. Traditional cost estimation methods do not have the capacity to fully utilize the existing tacit knowledge about past projects and their estimated and actual costs. Therefore, estimation methods tend to be slow [redacted] This leads to a significant financial impact on the preparation of a proposal for engineering projects. Due to the modern developments in computer technology and mathematical programming techniques, recently developed cost estimating approaches tend to use more complex methods and large volumes of data. These developments facilitated the emergence of Artificial Intelligence (AI) methods. In the literature, while there is a myriad of data-driven and AI-based cost estimation methods for contractor's, there are very limited studies on the development and application of similar methods for engineering consultancy firms. This research attempts to use the existing tacit knowledge in data about past projects to perform cost estimation on new projects by developing an accurate AI-based cost estimation method. Building on existing work on AI cost estimation methods, the research question is: How can an accurate AI cost estimation method be developed, to help engineering consultancy firms utilize the existing tacit knowledge that is captured in data to improve speed when estimating costs of engineering services in the tender phase?

Findings in the literature review revealed that artificial neural networks (ANNs) have the potential to overcome the previously described problem. Hereafter, the cost components that affect the costs of engineering services were identified by a literature review and interviews with experts. This led to the findings of 16 different variables that could potentially influence the proposal price for a tender. Eventually, the data of 132 projects were gathered using an online survey. Subsequently, a method was established to develop an ANN and to improve its performance. The method led to an optimal neural network consisting of a seven-neuron input layer, a four-neuron hidden layer that used sigmoid transfer functions and a linear single-neuron output layer. The best performing training algorithm was the Bayesian Regularization training algorithm. The most relevant input variables that influence the proposal price that were discovered are; project duration, number of project team members, number of disciplines, intensity, project phase, type of contract and scale of work. [redacted]

Eventually, the results showed that artificial neural networks (ANNs) can obtain a fairly accurate cost estimate quickly, even with small datasets. Whether the model is an improvement with regard to the pace of completion of tender could not be proven in this research, as no external validation was performed. However, In the interviews, some participant explained that they could provide the information for the variables that were determined for new projects within an hour after reading a RFQ.

With an average accuracy of 86,4% or mean absolute percentage error of 13,65% based on 12 individual test cases, the model is fairly accurate with respect to the accuracy that is obtained with the currently used estimation method. The work of Hyari et al. (2016) resembles the most with this research as it is the only research done towards developing an ANN for cost estimation of engineering services. The performance of the model that is described in this research is an improvement with regard to the work proposed by Hyari et al. (2016) as accuracy is higher and deviation in the prediction is lower. In their study, the average test performance of 71,8% or mean absolute percentage error of 28,2% was obtained, with a maximal error of an individual test result of 86,2%. Although the accuracy of the proposed model is relatively high compared to other researches, results from using the model in practice could lack in accuracy. The maximal error of an individual test result was 62,06%. Therefore, while the average accuracy of the testing results is relatively high, the deviation of the individual predictions is still high.

In addition, the training of a neural network is involved with stochastic elements, due to which every training run a different performance and different variance will emerge. To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. Based on the prediction of 100 different networks, The average MAPE is 61,73% with a standard deviation of 31,27%. Therefore, the more robust estimate of the MAPE of the model is larger compared to the final optimal model. This means, while the final model has reasonable accuracy, the model is perceived as very unstable. This is identified by taking the additional source of variance due to the stochastic nature of the model into account. Therefore, implementing this method in practice should be considered carefully and is not advised at this moment.

The developed AI cost estimation method has a high potential to grow. In order to successfully use the developed model in practice, several recommendations are suggested for further research. First, the model should be externally validated. This could be done by using the model alongside the currently used detailed estimation method. Subsequently, compare the prediction of the ANN model with the prediction of the current estimation method. When the model's accuracy is perceived as too low in order to apply it in practice, the model's accuracy can be improved by redeveloping it using more data. By saving relevant data in the databases, more and more data is collected over time. This data can then be used for developing a more accurate neural network. In addition, neural networks are accurate predictors however, the justification behind the prediction is very hard to do. By performing external validation trust can be built towards the neural network's abilities. This could also imply as a justification for the proposed price for management. However, bringing out a proposal based only on the ANN model still has a lot of challenges. This is an aspect that still needs some further research. For example, the following question can be asked: what are challenges regarding the adoption of a black box technology within an organization?

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial intelligence
ANNs	Artificial neural networks
Capex	Capital expenditures
CBR	Case-based reasoning
CERs	Cost estimating relationships
DCNs	Design change notices
E	Engineering
EPC	Engineering, Procurement, and Construction
EPCm	Engineering, Procurement and Construction Management
ES	Evolutionary systems
EXS	Expert system
FBM	Feature based method
HS	Hybrid systems
KBS	Knowledge-based systems
ML	Machine learning
MLR	Multiple linear regression
MRA	Multiple regression analysis
MSE	Mean squared error
OBS	Cost breakdown structure
RFQ	Request for quotation
SaaS	Software as a service
SVM	Support vector machine
UI	User Interface
VIF	Variance inflation factor
WBS	Work breakdown structure

1 INTRODUCTION

1.1 Research background

In a globally competitive world, with diminishing profit margins and decreasing market shares, the cost of delivering a service or product is one of the major criteria in decision making at the early stages of a building design process in the construction industry (Günaydin & Doğan, 2004). A cost estimate of capital expenditures (Capex) in the tendering phase of a project greatly influence planning, bidding, design, construction management and cost management (Arage & Dharwadkar, 2017). Decisions based on cost estimates commonly lead to resource allocation and other types of major commitments, which may have critical consequences. Cost estimates allow project managers to evaluate the feasibility of projects and control costs effectively. Furthermore, the estimate may influence the client's decision on whether or not to progress with the project (Ahiaga-Dagbui & Smith, 2012). In addition, for many clients completing the project within the predefined budget is a paramount determinant of client satisfaction. Therefore, inaccurate estimates of costs can result in a significant financial impact on a project and deteriorated relationships with clients.

Cost estimating practice
A cost estimate is generally established by a coordinating role of a tender manager supported by a technical expert (e.g. engineers and project managers) who is very experienced in a specific activity. Tender managers and technical experts who perform cost estimates are referred to as estimators. A cost estimation method can be described as the symbolic representations of a system that expresses the content of that system in terms of the factors which influence its costs (Kirkham, 2014). Currently, existing estimation methods require detailed information about the project and tend to be very time-consuming and therefore costly. In the tendering phase of a project, limited information is available to estimators for making a cost estimate. Due to the lack of information, they leverage their knowledge, experience, and make intuitive judgment calls in order to estimate project costs (Cheng, Tsai, & Sudjono, 2010). Estimators have different levels of experience, this leads to tangible differences in the accuracy of cost estimates. Estimation methods in the tender phase of a project need to be quick, realistic and reasonably accurate (H. J. Kim, Seo, & Hyun, 2012). However, this is very difficult in the absence of sufficient information and different levels of experience of estimators.

Contractor's vs engineering consultancy firms
Cost estimates can be made both for the costs of projects for contractors and the costs of projects for engineering consultancy firms (Zwaving, 2014). The contractor's role is generally to evaluate the client's needs and actually perform the work that is needed to realize and build the project. The consultant's role is to evaluate a client's needs and provide expert advice and opinion on what needs to be done, by providing services. Contractors have to consider all costs for building a project, on the other hand, engineering consultancy firms have to consider only the cost for their services. According to Elfaki, Alatawi, & Abushandi (2014), any construction cost estimation should be developed based on specific parameters such as the type of project, materials costs, likely design and scope changes, ground conditions, duration of the project, size of the project, type of client and tendering method. These can also be referred to as design and project specific variables. Contractors have to consider cost variables like materials costs, weather conditions, and ground conditions. In contrast to contractors, engineering consultancy firms do not have to consider these variables and are more inclined to consider a variable like the type of market (e.g. Oil & Gas, Infrastructure, Industry, and Utilities & Environment). Engineering firms tend to operate in several different markets and contractors usually focus more on one particular market or activity. Operating in several different markets is associated with other types of risks than operating in one particular market (e.g. level of detail of designs and regulations). In general, the characteristics of cost estimations are different for contractors and engineering consultancy firms (Zwaving, 2014).

Traditional cost estimation methods
Various estimation methods and techniques have been proposed in the literature, for instance, traditional detailed estimating, comparative estimating, probabilistic estimating and parametric estimating. Detailed estimation methods tend to be very time-consuming in conducting an estimate and are associated with high costs. Furthermore, a new estimate should be established for every new project. With comparative estimating the accuracy is very limited due to the fact that normalization of a past project is required by an expert, this can lead to a subjective appreciation of the data. With probabilistic estimation methods for each cost component a cost distribution and correlation should be identified, this is considered a difficult process and is not always performed correctly. Parametric estimation methods can make use of a linear relationship between final cost and project specific variables based on previous projects. The assumption about a linear relationship between costs and project specific variables such as project

size, type of work, type of contract, type of client is questionable (Günaydin & Doğan, 2004). For example, when a client has relatively high demands, this could be measured on a qualitative scale. However, we cannot measure how much this influences the costs, be determining a linear relationship. Many studies tried to investigate the establishment of non-linear relationships within traditional methods. These studies generated higher-level predictability depending on the quality of the underlying data source and the sophisticated statistical techniques employed to build the model (Chou, Yang, & Chong, 2009). However, due to a large number of significant variables defining non-linear relationships or even linear relationships turns out to be very difficult (Cheng et al., 2010). For example, using only 4 different parameters for a project and considering three alternative values for each, and varying one at a time will produce 81 different project solutions or alternatives (Ahiaga-Dagbui & Smith, 2012). Therefore, while there usually is a rich record of estimates and the actual costs for previous projects, this implicit knowledge is usually ignored or under-utilized as a result of the capabilities of these traditional cost estimation methods.

Artificial intelligence (AI) methods
Due to the modern developments in computer technology and mathematical programming techniques, recently developed cost estimating approaches tend to use more complex methods and large volumes of data. These developments facilitated the emergence of Artificial Intelligence (AI) methods, which allow investigating multi- and non-linear relationships between final costs and design variables (Günaydin & Doğan, 2004). In addition, researchers claim that even with limited information it is possible to obtain a fairly accurate cost estimate quickly (Günaydin & Doğan, 2004). Current methods include machine-learning (ML), knowledge-based systems (KBS), evolutionary systems (ES) and hybrid systems (HS) (Elfaki et al., 2014). AI methods use large volumes of data that are stored from previous tenders and identifies patterns or relationships within these datasets by a self-learning process. The identified relationships are not prone to the subjectivity of estimators, and the use of AI methods minimizes the impact on the accuracy of an estimate that is caused by the different levels of experience that estimators have. These AI methods do use the rich record of estimates and actual costs that are known for previous projects and therefore do utilize the implicit knowledge on project execution.

Literature solutions
In the literature, while there is a myriad of data-driven and AI-based cost estimation methods for contractor's, there are very limited studies on the development and application of similar methods for engineering consultancy firms. More specifically, a lot of literature is available about the relevant design and project-specific factors that influence costs for contractors in the construction industry. However, there are few studies that contributed to establishing a benchmark for relevant design and project specific variables that are used in utilizing tacit knowledge in data for engineering consultancy firms.

1.2 Problem statement

The expected rate or pace of the completion of tenders which engineering consultancy firms need to perform is increasing. Traditional cost estimation methods used by engineering consultancy firms do not have the capacity to fully utilize the existing tacit knowledge about past projects and their estimated and actual costs. Therefore, estimation methods tend to be slow [redacted] This leads to a significant financial impact on the preparation of a proposal for engineering projects. Furthermore, the existing literature does not cover the specific solutions to overcome this problem for engineering consultancy firms.

1.3 Research goal

The aim of this research is to use the existing tacit knowledge in data about past projects to perform cost estimation on new projects by developing an accurate AI-based cost estimation method. By doing so, increasing the pace of preliminary cost estimation in engineering consultancy firms is ought to be achieved. The developed method should be able to estimate a preliminary proposal price as accurate and as quickly as possible

1.4 Research questions

Based on the problem statement and research objective the following research question is established: How can an accurate AI cost estimation method be developed, to help engineering consultancy firms utilize the existing tacit knowledge that is captured in data to improve speed when estimating costs of engineering services in the tender phase? In order to answer the main research question, the following sub-questions are identified:

1. What are the cost estimation methods that are commonly used by engineering consultancy firms and what are problems regarding these methods?
2. What modern AI-based cost estimation method can potentially overcome the problems of the current cost estimation methods?
3. Which preliminary cost components are relevant in establishing an AI-method, what implicit data is available and how can the required data be collected?
4. How can a cost estimation method that fits the problem be established and how does it perform?
5. In what way is the new modern estimation method an improvement with regard to traditional cost estimation methods?
6. What are the important weaknesses and limitations of the developed method, what conclusions can be drawn and what recommendations can be made to improve the use of the developed method?

1.5 Research client

One of the firms that deals with the research problem is Bilfinger Tebodin, their case act as a context for the research that is conducted. Bilfinger Tebodin is an international consulting and engineering firm owned by the German construction company Bilfinger. Bilfinger Tebodin comprises approximately 3,200 employees in seventeen countries. Offices can be found in Europe and the Middle East. The services offered include consultancy, design and engineering, procurement and construction and project management. The company is active in markets such as Oil & Gas, Infrastructure, Industry, Utilities & Environment, Property and Health & Nutrition. The company is well known for their knowledge of the different markets, vision on current developments, passion for technology and integrated consultancy and engineering services.

The offices in the Netherlands are part of the North West Europe network of Bilfinger Tebodin. The projects that are carried out can consist of activities like design and engineering, project management, procurement, construction management, and consultancy. The design and engineering activities are performed for four different project phases and contribute to the establishment of four different designs namely masterplan's, conceptual designs, basic designs, and detailed designs. All the activities contribute to either brown-field or green-field developments. Brown-field developments consist of expansions and modifications of client's assets. Green-field developments contribute to the creation of new assets for clients. In the specific case of Bilfinger Tebodin, the expected number of tenders is increasing and the available time to complete these tenders is decreasing. The current cost estimation method is very time consuming and therefore costly. Furthermore, the method used requires a well-known product and project specification in order to create a reliable estimate. Due to these facts, the estimation method tends to be slow [redacted]

1.6 Research strategy

The research framework (see Figure 1-1 below) gives an insight into the methodology and strategy that is used during this research in order to find answers to the research questions. The main research question is answered by giving answers to six sub-questions. These sub-questions are numbered and can be found in the corresponding phases in the figure below. To answer the sub-questions several steps are executed, these actions are described per phase and are elaborated below. Furthermore, the research framework shows the phases and their corresponding chapters within this report. The first two research questions are answered in chapter 2 which is labelled as a literature review. The third, fourth, and fifth research questions are answered in chapter 3 and 4, which are respectively labelled as the proposed method and results. Lastly, the final research question is answered in chapter 5 and 6 which are respectively labelled as discussion and conclusions.

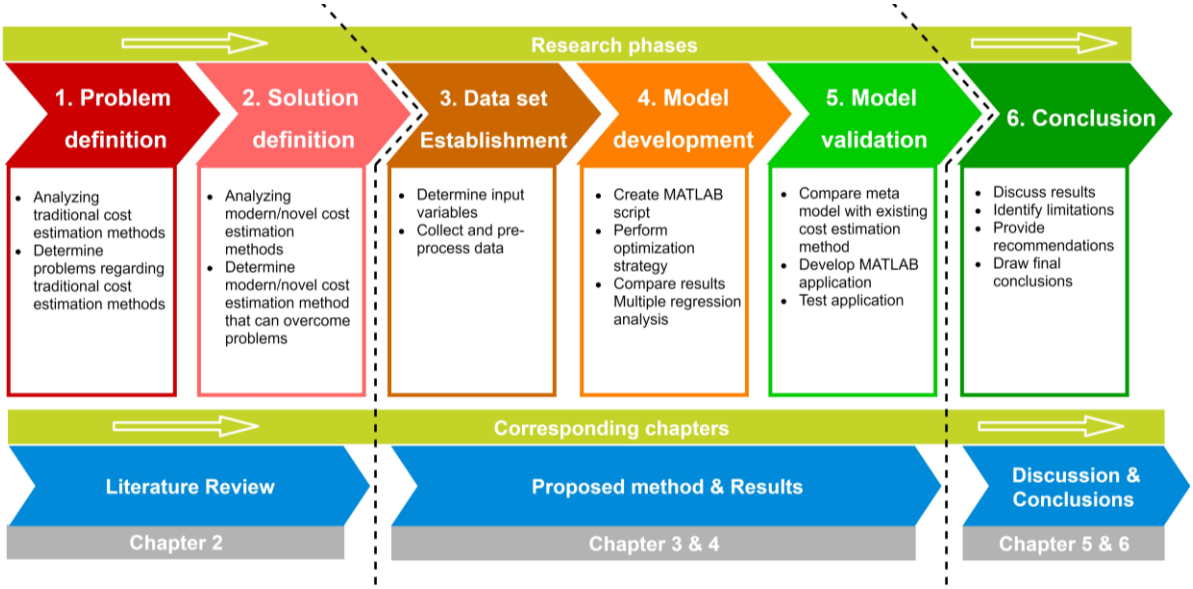


Figure 1-1. Research framework

Phase 1: Problem definition
First of all, it is important to know and identify what the current traditional cost estimation methods used by engineering consultancy firms are. For these methods, the pros and cons are identified by reviewing the literature. Furthermore, a clarification of the incapability of the currently used traditional cost estimation methods is provided. In this part, it becomes clear why the most used traditional methods are ineffective. In addition, the problems regarding the cost estimation method that is used by the research client need to be defined. Therefore, the current cost estimation process was analysed and described by reviewing the quality management systems.

Phase 2: Solution definition
In order to overcome the problems regarding the current cost estimation methods, possible solutions to these problems should be identified. This was done by conducting a literature review that focuses on modern cost estimation methods and their application in general. These modern cost estimation methods are not yet broadly used in cost estimating practice in engineering consultancy firms. By reviewing the benefits and drawbacks that are inherent to these cost estimation methods, a trade-off could be made between the available methods. This trade-off considers the benefits and drawbacks and weighs them off in order to see what method best fits the problem at hand and best fits the available data structure. Eventually, the best fitting cost estimation method was used in this research.

Phase 3: Dataset establishment
The research explores the possibilities to utilize tacit knowledge about project execution in existing data of engineering consultancy firms that can be used for establishing cost estimates. It does so by using the case of Bilfinger Tebodin as a context for the research that is conducted. Therefore, it is important to get insight into the available data. In order to achieve this, an analysis was performed. This analysis consisted of reviewing the software tools that are used to estimate the costs of services, reviewing used databases, reviewing quality management systems and performing unstructured interviews with relevant stakeholders. Eventually, the analysis provided knowledge about what cost relevant data is available and what data is used in the estimation.

To reach the research objective a cost estimation method should help utilize internal knowledge about project execution in existing data of engineering consultancy firms. Therefore, it is important to evaluate which specific data-criteria are relevant and should be used as input for the method. This means the relevant design and project-specific factors that are used for cost estimation for engineering consultancy firms should be identified. This was achieved by conducting a literature study about relevant design and project-specific factors that influence the costs of engineering services. In addition, semi-structured face-to-face interviews with experts were carried out to identify factors that are specifically relevant for consultancy firms. The requirements regarding the output of the method were determined based on the availability of data and in consultation with stakeholders and experts in the field. It was decided that the output of the model should be based on the proposal prices that are established after a tender.

Subsequently, when the required input and output criteria were known the data was gathered from various sources. Not all the data could be extracted from the databases. Therefore, a survey was set up to gather the data by asking relevant project managers and tender manager to provide information about projects they were involved in. When the data was gathered, a database with only the relevant data criteria was established. Then, the data was cleaned to have homogeneity. This was done because the data could have blank cells or divergent values. For this research, extreme values and blank values were either re-coded or deleted from the dataset and missing values replaced with the mean or mode of the dataset. Input variables can be of a qualitative nature or quantitative nature. The method used only processes quantitative data and therefore qualitative data were categorized into sub-variables (e.g. Good, Moderate, Poor, Not Applicable). These sub-variables were then processed into quantitative data by defining a corresponding numerical scale. The last action that was required in the establishment of the dataset was assigning the proposal prices of the real projects to the different project input datasets.

Phase 4: Model development
The AI cost estimation method was established by creating a model in the software MATLAB. This was done by importing the database that contains the input dataset and the output dataset. A code was written to import the data and process the data. In this code, the right settings and structure of the algorithm were created. Subsequently, the neural network was trained and the performance was analysed. To improve the performance of the model, an optimization strategy was established. This strategy consisted of three iterative processes that contributed to the improvement of the model. For all the three iterative processes the growing technique was used, which is a technique to determine the best network architecture, alluding to the number of hidden neurons in the hidden layer. At first, the best performing training algorithm was identified by training and testing three different training algorithms namely; Levenberg-Marquardt algorithm, Bayesian Regularization algorithm, and the Resilient backpropagation algorithm. The second iterative process determined the most important input variables that explained the dataset. This was done by calculating the relative importance of the variables and consecutive exclude the least important input variable for every iteration. The relative importance of the variables was determined by three methods namely; connection weight algorithm, multiple linear regression analysis and expert opinion. The third and final iterative process evaluated the influence of project value ranges on the performance of the model. Finally, a model with the best performing training algorithm, input variables, project value range, and network architecture was identified.

Phase 5: Model validation
The fifth phase of the research focused on the internal validation of the developed method. The accuracy of the method was determined by comparing method output in terms of cost estimates with proposal costs of real-world projects. This was done by using a split-sample method, in where the dataset was split into a training set and a test set. The internal validation provided insights into how the model will perform outside the training sample. Therefore, a feeling is acquired for the generalization of the model. In addition, a common source of variance in a final model is the noise in the training data and the use of randomness in the training phase. The training of a neural network is involved with two stochastic elements, due to which every training run a different performance will emerge. The first stochastic element regards to the random initialization of the weights and the second stochastic element regards to the random division of the datasets. To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. This was done by training a model several times and evaluate the variance that is introduced by the stochastic elements. Furthermore, the neural network algorithm that is developed in MATLAB is transformed into a function, that is connected to a stand-alone application. This application has a user interface in which the input variables for new projects can be entered. The application can then provide a prediction of the costs and can be used in practice for new tenders.

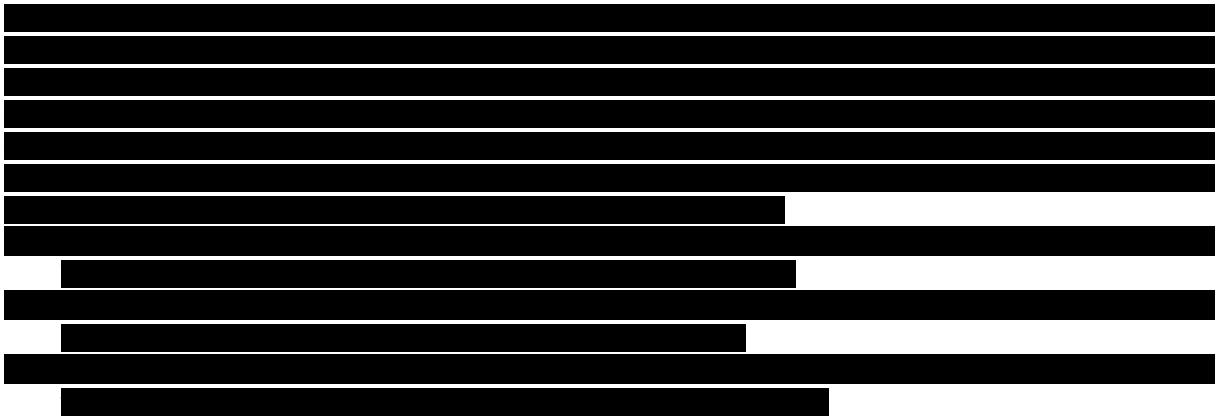
Phase 6: Conclusion
The last phase of the research is focused on analysing and interpreting results to come to conclusions and recommendations. In this phase, the results are discussed and an evaluation is carried out to determine the weaknesses and limitations of the newly developed method. By doing so, awareness is created about the use of the developed method and the risks that are involved with the use of the method. In order to overcome the weaknesses and limitations and improve the use of the developed method, recommendations are provided for implementation of the method and for future research. Lastly, when the results are discussed, the limitations are known and the recommendations are known, the overall conclusion of the research is provided.

1.7 Relevance

As explained in the research background this research focuses on cost estimation methodology of engineering consultancy firms. In the literature, a lot of cost estimation methods that utilize the implicit internal knowledge that exists in the rich records of estimates and actual costs within the contractor's context are researched. However, few researchers make contributions towards cost estimation methods that utilize the internal knowledge for engineering consultancy firms. More specifically, a substantial amount of literature is available about the relevant design and project-specific factors for contractors in the construction industry. However, there is no benchmark for relevant design and project specific variables for engineering consultancy firms that deliver engineering services.

Research that contributes towards AI-based cost estimation methods of services for engineering consultancy firms within the construction industry is, to the best of knowledge, only done once and can still be defined as a scientific novelty. There are several additional contributions this research ought to make to scientific literature, these are the following:

- Providing an overview of both AI and traditional cost estimation methods that can be used for estimating the costs of engineering services.
- Exploring the potential of AI-based cost estimation methods towards estimating the costs of engineering services.
- Identifying the effect that AI-based cost estimation methods have on the accuracy and duration of estimating the cost of engineering services.
- Provide a benchmark for relevant design and project-specific factors that influence cost estimates of services for engineering consultancy firms.
- Provide knowledge towards the use of AI-based solutions towards the preparation of tenders within the construction industry.



2 LITERATURE REVIEW

2.1 Traditional cost estimation methods

In this chapter, an answer to the first research question is provided. This answer is established by performing a literature review on work that already has been carried out by other academics in the area of cost estimating of engineering services. In the literature review, the research areas that are relevant are identified. In addition, the current understanding of these areas is identified. Furthermore, insights are provided in the opposing views that are identified within the scientific knowledge in the field.

The literature provides comprehensive knowledge of cost estimation methods for construction projects. However, few researchers make contributions toward cost estimation methods for engineering services. Nevertheless, many of the methods used in estimating costs for construction projects can also be used for estimating the costs of engineering services (Zwaving, 2014). Traditional methods that are identified can broadly be divided into parametric, detailed, comparative and probabilistic estimating (Table 2-1). In this section, the different traditional cost estimation methods of engineering services are described and the importance of these methods is elaborated. Subsequently, the pros and cons of these different methods are identified.

Table 2-1. Literature sources of traditional estimation methods

Method category:	Estimation method:	Sources:
Traditional estimation method	Parametric, Feature-based or Multiple regression analysis estimating	(Chou et al., 2009; Gao, 2009; Hamaker, 1995; NASA Executive Cost Analysis Steering Group, 2015; Zwaving, 2014)
"	Detailed, Bottom-up or Analytical estimating	(Gao, 2009; NASA Executive Cost Analysis Steering Group, 2015; Zwaving, 2014)
"	Comparative or Analogy estimating	(Burke, 2009; Lester, 2017; NASA Executive Cost Analysis Steering Group, 2015; Zwaving, 2014)
"	Probabilistic or stochastic estimating	(Elkjaer, 2000; NASA Executive Cost Analysis Steering Group, 2015; Zwaving, 2014)

2.1.1 Parametric estimating

The parametric estimating technique is also known as feature based method (FBM) or multiple regression analysis (MRA) (Chou et al., 2009). In the parametric method, a statistical relationship is developed between historical costs and project attributes by performing a regression analysis. These project attributes or variables usually consist of program, physical, and performance characteristics (Gao, 2009). For example, variables could be time, location, currency, productivity and complexity. A parametric estimation is obtained by identifying these relationships that are also known as Cost Estimating Relationships (CERs) and applying an algorithm to determine an approximation of the total project costs (Kwak & Watson, 2005). The variables that are used in a parametric estimate should be the cost drivers of the project. The assumption that is made is that the variables that affected cost in the past will continue to affect future costs. The use of a parametric method requires access to historical data that can be used to determine the cost drivers and the relevant CERs. The parametric CERs can then be used for cost estimates for future projects based on the specific characteristics of the project.

The major advantage of using a parametric methodology is that the estimate can usually be conducted quickly and be easily replicated (NASA Executive Cost Analysis Steering Group, 2015). Furthermore, a parametric estimate eliminates the reliance on opinion through the use of actual observations. A disadvantage regarding parametric estimating is the fact that the CERs should be continually revisited, in order to assure that they are in line with the current relationship between project attributes and costs. Furthermore, CERs should be correctly and precisely documented as serious estimating errors could occur if the CERs are improperly used (Gao, 2009). In addition, Hamaker (1995) argues that most CERs are linear relationships, meaning that there is a single value of the independent variable associated with a cost driver. Many studies have explored non-linear relationships and generated higher-level predictability depending on the quality of the underlying data source and the sophisticated

statistical techniques employed to build the model (Chou et al., 2009). Performing the correct statistical techniques that are needed to build a quality model is considered as difficult.

2.1.2 Detailed estimating

The detailed estimation method is also often called a bottom-up or analytical estimation method. This method produces a detailed project cost estimate that is computed by estimating the duration of every activity that is carried out in a project (NASA Executive Cost Analysis Steering Group, 2015). This is done by first establishing a Work Breakdown Structure (WBS) and computing the work effort of a WBS element. Subsequently, the costs per activity are calculated and connected to the WBS elements resulting in the establishment of a Cost Breakdown Structure (CBS). The establishment of a WBS and the estimation of the work effort is generally done by a technical person who is very experienced (e.g. engineers and project managers) in a specific activity.

A big advantage of the detailed estimation method is the ability to determine exactly what the estimate include and whether anything was overlooked (Gao, 2009). In addition, the method provides insights into the major cost contributors to the project. Furthermore, the activities that are distinguished in a project are usually reoccurring and can be reused in future projects. There are also several disadvantages regarding the detailed estimation method. The first is that the process of executing a detailed estimate can be very time consuming and therefore costly. Another disadvantage is the fact that a new estimate must be established for every new project. Estimates of certain activities that are reoccurring can be taken from previous projects but must be integrated into the context of the new estimate. Furthermore, the product and project specifications must be well known and stable in order to create a reliable estimate. When the product and project specifications change over time these changes must be reflected in the estimate on a continuous basis. Lastly, small errors can grow into larger errors during the summation of the different WBS elements.

2.1.3 Comparative estimating

The comparative estimation method or also known as the analogy cost estimation method uses the cost of similar projects, considers the differences and estimates the cost of the new project (NASA Executive Cost Analysis Steering Group, 2015). This method is based on the costs of a simplified schedule of major activities that were used on previous similar projects (Lester, 2017). It is based on the costs of major cost components that were used on previous similar projects for which recent experience is available. A comparative estimate is generally used to investigate the feasibility of the project and provides information about whether to proceed with the project within the defined boundaries (Burke, 2009). Besides that, the analogous approach is also used when attempting to estimate a generic system with little available definitions.

One of the biggest advantages of the comparative estimation method is that it is extremely quick in completing an estimate. It can be accurate if there are minor deviations with respect to the data from previous projects on which the estimate is based. The reasoning behind the established estimate is readily understood by everyone involved. However, it can also be very difficult to identify the appropriate project that has similar aspects to compare it with the new project. The process relies on extrapolation and expert judgment for the adjustment of the factors. Therefore, the requirement of normalization can lead to a subjective appreciation of the data and can influence the accuracy of the estimate. Gao et al. (2009) argue that adjustments of the factors should be made as objectively as possible, using factors that represent differences in size, performance, technology or complexity.

2.1.4 Probabilistic estimating

The probabilistic estimation method presents a probabilistic estimating range that cannot be offered in the other traditional estimation methods that are mentioned above (Chou et al., 2009). The method uses probability distributions for one or more parameters as input for the cost estimate (Zwaving, 2014). It focuses on the risks and uncertainties involved in the project and attempts to quantify the project cost variability. The method gives insight into the change of exceeding a particular cost in the range of possible costs, how much the cost could overrun and uncertainties and how they drive costs. According to NASA Executive Cost Analysis Steering Group (2015), a probabilistic estimation method allows to more effectively communicate the impact of changes to planned or requested resources by providing quantified effects on the probability of meeting planned cost and schedule baselines. Furthermore, at the proposal stage, the design and demands are still relatively unclear. At this stage, it is sensible to consider uncertainties and to use probabilistic range estimation rather than a single point or deterministic estimation (Elkjaer, 2000).

The probability distribution is crucial in simulation modelling and occasionally influences output accuracy (Chou et al., 2009). A probability distribution is a statistical function that describes all the possible values and likelihoods that a random variable can take with a given range (Zwaving, 2014). Based on a predetermined confidence level a probability density distribution of the total cost can be established. Therefore, an advantage is that the probability of cost overrun is insightful. This lead to a substantiated accuracy of the estimate. Two main challenges task exist when using this cost estimation method. First, for each cost component a cost distribution should be identified (Chou et al., 2009). Second, the correlation between cost components must be identified. If this is not done correctly, the reliability of the estimates can be questionable.

2.2 The incapability of traditional methods

In engineering consultancy firms there are some commonly used practices and one of them is the cost estimation method. Every company has its own specific system to perform this method, however, the general principles that cover the method are somehow the same. In this section, the traditional cost estimation methods and the main problems regarding these methods will be elaborated more in depth. When enough information about a project is available engineering consultancy firms like Fluor and Bilfinger Tebodin commonly use the detailed estimation method to determine the costs of a project. When insufficient information about a project is available a comparative estimation method is used to estimate the costs of a project. Parametric and probabilistic estimation methods are methods that are less commonly used in engineering consultancy firms, however, the problems regarding these methods are also described below.

The detailed estimation method can be very time consuming and therefore costly. For this method, the product and project specifications must be well known and stable in order to create a reliable estimate. This information is not always available in the early stages of a project and therefore an accurate estimate is not always achievable within the available tender time frame. Furthermore, a new estimate needs to be established for every new project and can only use a limited amount of internal tacit knowledge of previous projects in new project estimates. This limited amount refers to the reoccurring activities in similar previous projects that can be used in the new WBS. Therefore low utilization of the tacit knowledge in data is performed with the use of this method. Therefore, due to the increased expected rate at which tenders need to be performed and that it does not utilize the internal tacit knowledge in data this method is not suitable for the future anymore. Because, in time, using this method can have an impact on the competitiveness of a company.

With the comparative cost estimation method, an estimate can be established very quickly even without sufficient project information. However, this estimate is based on estimators knowledge, experience, and intuitive judgment calls (Cheng et al., 2010). Due to the fact that estimators have different levels of experience, this leads to tangible differences in the accuracy of cost estimates. Accuracy is important as a cost estimate in the tendering phase of a project greatly influence planning, bidding, design, construction management, and cost management. Furthermore, the estimate may influence the client's decision on whether or not to progress with the project. Due to the tangible differences in the accuracy of cost estimates the comparative method is not suitable to establish sufficient accurate estimates. Also, this method does not use the tacit knowledge that is available in data in order to learn from the past.

Parametric estimating has the capacity to utilize existing knowledge of project execution into new estimates, however, most CERs are linear relationships and non-linear CERs are very hard to establish. It is questionable whether relationships between cost factors and final costs are linear and these relationships are more likely to be non-linear. Furthermore, CERs should be continually revisited to assure that they are in line with the current relationship between project attributes and costs. Therefore, the whole process of establishing CERs is a continuous and time-consuming activity. This method is not appropriate in a world that accelerates significantly. Based on these facts the parametric estimation methodology is considered to not be an appropriate solution for the specific research problem.

The probabilistic cost estimation method makes use of probabilistic cost distributions for each cost component. This process is considered hard to achieve and these probabilistic cost distributions should continually be revised. Therefore, the process of establishing cost distributions should also be carried out on a regular basis. In addition, the probabilistic cost estimation method should always be performed based on either the parametric estimation method, the detailed estimation method or the comparative estimation method. Therefore depending on the method, the time it takes to perform an estimate can be long or short. Based on these facts the probabilistic method is not appropriate for the problem at hand. In order to provide a clear image of the pros and cons of the different cost

estimation method, all the strengths, and weaknesses of the different traditional cost estimation methods are summarized in Table 2-2 below.

Table 2-2. Strengths, weaknesses, and requirements for distinguished cost estimation methods

Estimation method	Strengths	Weaknesses	Requirements
Parametric estimating	• Quick and accurate way to estimate costs • An estimate can be easily replicated • Estimate eliminates the reliance on opinion through the use of actual observations • Reducing the cost of preparing project proposals	• Documentation of Cost Estimating Relationships (CERs) can be difficult • Improper use of CERs can lead to serious estimating errors • CERs should be continually revisited • Most CERs are a linear relationship and non-linear CERs are very hard to establish	• Historical data for statistical analysis • Statistical software • Sophisticated statistical knowledge
Detailed estimating	• Very high accuracy of the estimate • Ability to determine exactly what the estimate include and whether anything was overlooked • Enables insights into the major cost contributors to the project • Some activities that are estimated can be reused in future projects	• Project's scope must be determined and understood considerably • Very time consuming to conduct the estimate • High costs to establish the estimate • A new estimate for every project • Small errors can grow into larger errors during the summation of the different WBS elements • Estimating depends on the availability of experts	• Work breakdown structure • Man-hour estimates • Experts for estimating man-hours • Collaboration between employees • Sufficient available information about the project.
Comparative estimating	• Very quick in estimating costs • Accurate if the project is similar to a project that has been carried out • Doable without complete scope understanding • The reasoning behind the established estimate is readily understood by everyone involved	• Accuracy is very limited • Normalization required which lead to a subjective appreciation of the data • Depends on the similarity of finished projects • Hard to identify a similar project	• Knowledge or data of existing comparative projects • Comparison factors
Probabilistic estimating	• Insight in the probability of cost overrun • The substantiated accuracy of the estimate	• For each cost component, a cost distribution should be identified, which can be difficult • The correlation between cost components must be identified, which can be difficult • Probability distributions should continually be revised.	• Historical data in order to establish a probability distribution of cost components • Statistical software • Sophisticated statistical knowledge

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

2.4 Artificial intelligence estimation methods

Now that the problems and incapability's of traditional cost estimation methods and the cost estimation method used by the research client are known, possible solutions to overcome the research problem can be distinguished. In order to achieve this, modern and novel cost estimation methods that have the ability to fully utilize the tacit knowledge that exists in data are reviewed in the literature. This review contributed to the establishment of describing four different modern AI methods that regard solving cost-related problems. This section provides an answer to the second research question of this research.

Due to the modern developments in computer technology and mathematical programming techniques, recently developed cost estimating approaches tend to use more complex methods and large volumes of data. The developments in mathematical programming techniques facilitated the emergence of Artificial Intelligence (AI) and AI tools. AI tools allow investigating multi- and non-linear relationships between final costs and design variables (Günaydin & Doğan, 2004). Elfaki et al., (2014) distinguished four different state-of-the-art AI-based approaches that are machine-learning (ML), knowledge-based systems (KBS), evolutionary systems (ES) and hybrid systems (HS) (Table 2-3) These four different AI cost estimation methods will now be described.

Table 2-3. Literature sources of AI estimation methods

Method category:	Estimation method:	Sources:
AI estimation methods	Machine-learning (ML)	(Bosscha, 2016; Elfaki et al., 2014; Günaydin & Doğan, 2004; Petroutsatou, Georgopoulos, Lambropoulos, & Pantouvakis, 2012; Rafiq, Bugmann, & Easterbrook, 2001; Son, Kim, & Kim, 2012)
"	Knowledge-based systems (KBS)	(Elfaki et al., 2014; K. J. Kim & Kim, 2010; Tripathi, 2011)
"	Evolutionary systems (ES)	(Elfaki et al., 2014; Mirjalili, 2018)
"	Hybrid Systems (HS)	(Cheng et al., 2010; Elfaki et al., 2014)

2.4.1 Machine-learning

ML systems have been defined as a system that can self-learn from data that it deals with (Elfaki et al., 2014). The main benefits regarding machine-learning are the ability to deal with uncertainty, the ability to work with incomplete data, and the ability to judge new cases based on acquired experiences from similar cases. Furthermore, they can investigate the multi- and non-linear relationship between cost parameters and are self-learning. The main disadvantage regarding ML is the lack of technical justification, that is, the causes behind the decision are not known. This phenomenon is also known as a black box decision. Machine-learning systems can be divided into two

main ML techniques. The first is an artificial neural network (ANN) and the second is the support vector machine (SVM).

An artificial neural network (ANN) is a computational model based on the structure of biological neural networks. By providing an artificial neural network with input datasets with known corresponding output values, the neural network can train itself and can learn from the data that is available. ANNs can solve problems without the benefits of an expert and they can seek patterns in data that are not obvious (Ahiaga-Dagbui & Smith, 2012). The ability of neural networks to learn gives an advantage in solving complex problems whose analytic or numerical solutions are hard to obtain (Rafiq et al., 2001).

Support vector machine systems are a novel and powerful learning method based on statistical learning theory. The SVR model does not depend on the dimensionality of the input layer, it has a relatively high performance with smaller datasets compared to the ANN model (Son et al., 2012). Therefore, it has an advantage over ANNs when only small datasets are available. However, the ability for SVR systems to deal with multi- or non-linear relationships between cost parameters is none existing. This means only simpler relationships that consist of linear relationships can be identified.

2.4.2 Knowledge-based systems

Elfaki et al. (2014) describe knowledge-based systems (KBS) as any technique that used logical rules for deducing the required conclusions. The goal of a KBS is to capture the knowledge of a human expert from a specific domain and code this in a computer in such a way that the knowledge of the expert is available to a less experienced user (Tripathi, 2011). The main advantages of KBS are the ability to justify any result and the fact that it is easy to develop a KBS. However, the disadvantages are that it is difficult for a KBS to self-learn and the initial rule establishing process is very time-consuming. Here commonly used techniques within KBS are expert system and case-based reasoning. Case-based reasoning (CBR) is the process of retrieving previous cases similar to a new problem, solving the new problem by adapting previously determined solution of the similar previous cases and storing the new successful solution for future use (Kim & Kim, 2010). Expert systems (EXS) is a computer programme that simulates the judgment and behaviour of a human that has expert knowledge and experience in a particular field (Tripathi, 2011). The programme achieves this by reasoning through bodies of knowledge, represented mainly as if-then rules. When designing an expert system, one should always keep in mind two parts: a knowledge base, which is a database containing facts, rules, relations etc. and an inference engine which interprets the knowledge's and controls the problem-solving procedure according to a predefined strategy (Engelmore & Feigenbaum, 1993).

2.4.3 Evolutionary systems

Evolutionary systems (ES) involve techniques implementing mechanisms inspired by biological evolution such as reproduction, mutation, recombination, natural selection and survival of the fittest. The system is concerned with continuous optimization with heuristics (Elfaki et al., 2014). ES are used as an optimization tool where there are many solutions but the right solution is not known. In this method, an initial set of candidate solutions to a specific problem is generated and iteratively updated. Each new generation is produced by stochastically removing less desired solutions and introducing small random changes. ES are also mostly population-based paradigms. This means they iteratively evaluate and improve a set of solutions instead of a single solution (Mirjalili, 2018). The main limitation regarding the use of evolutionary systems is the fact that ES are generated based on specific heuristics and are therefore difficult to generalize. Furthermore, it is difficult for an evolutionary system to self-learn.

2.4.4 Hybrid systems

Hybrid systems (HS) consist of a combination of different techniques in order to solve a specific problem. Usually, specific techniques have certain limitations, and combining two or more different techniques in one can allow overcoming these individual limitations. For example, Cheng et al. (2010) proposed a hybrid system for construction costs index modelling. In this case, the model is composed of support vector machine aspects and evolutionary system aspects. The implementation of HS could be a problem due to the unavailability of computational tools that could support the implementation. Furthermore, extensive knowledge of different techniques is required to achieve good results. Therefore, when implementing a hybrid system the initial effort to establish the method is significantly higher. However, when executed correctly a hybrid model can give better result compared to individual methods.

2.5 The appropriate cost estimation method

In this chapter, the pros, and cons of the different cost estimation methods are reviewed. In order to get a clear view of which cost estimation method is the most appropriate to reach the research objective, in this section a comparison is carried out. In addition, a table (Table 2-4) is established to summarize the benefits and drawback of different modern cost estimation methods. Knowledge-based systems (KBS) are easy to develop, however, the initial rule establishing process is very time-consuming. Furthermore, the implicit tactic knowledge that is needed in order to establish an accurate cost estimate is too complex to translate into easy and clear sets of if-then rules. In addition, KBS can only be used for classification problems. Whenever the output variable described some categories or discrete classes than there is a classification problem, this is not the case in this research. Therefore, knowledge-based systems are not valid for the specific problem at hand. Evolutionary systems are more appropriate for problems where there is an initial set of candidate solutions available. Furthermore, problems regarding population-based paradigms are more suitable when using this method. Therefore evolutionary systems are not relevant. Hybrid systems are interesting in the way that they have the ability to overcome the limitations of individual methods. However, the use of a specific method requires extensive knowledge of that particular technique. Furthermore, different computational tools are required in order to achieve a solid method. Therefore this method is out of scope in this research.

The method that is distinguished as the most appropriate method to overcome the existing problem is machine learning and in particular artificial neural networks (ANNs). ANNs have the ability to self-learn which saves a lot of time in the establishment and revision of the method. The method learns from existing data and determines the correlation between cost factors and project cost by a predefined algorithm. ANNs can identify non-linear relationships between cost factors and project cost with no additional effort. Once the method is established, an estimate or prediction of the costs of a project can be generated very quickly. Kim, An, & Kang, (2004) analyzed three cost estimating models namely artificial neural networks (ANNs), multiple regression analysis (MRA) and a case-based reasoning system (CBR) and concluded that ANNs worked more accurately then MRA and CBR estimating models. Furthermore, according to Cheng et al. (2010) ANNs represent the most frequently applied approach in estimating the duration and costs of construction projects during the preliminary design stage. With an ANN model, it is possible to obtain a fairly accurate prediction, even when sufficient information is not available in the early stages of the design process (Günaydin & Doğan, 2004). Furthermore, the company that acts as a context for this research (Bilfinger Tebodin) claim they have a significant amount of data available. This is one of the requirements when developing a machine learning algorithm. Based on these facts the machine learning cost estimation method has the highest potential to overcome the research problem.

Table 2-4. Strengths, weaknesses, and requirements for distinguished modern cost estimation methods

Estimation method	Strengths	Weaknesses	Requirements
Machine Learning	• Self-learning ability • Ability to integrate and deal with uncertainty • Can be retrained with new data easily. • Ability to judge new project costs based on acquired experience from previous projects costs. • Quick in estimating costs • Very accurate in estimating costs • Identify multi- and non-linear relationships between cost parameters	• Lack of technical justification • Black box decision • Needs a large amount of data • Hard to determine input parameters	• Large data sets • A pre-defined set of variables for the input layer • Corresponding target values for variables in the input layer. • Statistical software • Statistical knowledge
Knowledge-based systems	• Ability to justify any result • Easy to develop a KBS • Knowledge preservation	• Difficult for a KBS to self-learn • The initial rule establishing process is very time-consuming	• Set of If-then rules. • Expert(s) with extensive knowledge • An inference engine
Evolutionary systems	• Ability to remove less desired solutions • Good for population-based paradigms • Optimization of a set of solutions	• Is generated based on specific heuristics and are therefore difficult to generalize.	• The initial set of candidate solutions • Population-based paradigms
Hybrid systems	• Ability to overcome limitations of individual methods • Benefiting from several advantages of different methods	• Hard to implement due to the unavailability of computational tools • Extensive knowledge of different techniques is required to achieve good results.	• Extensive knowledge of different techniques • Computational tools that are able to handle techniques.

Despite the broad attention and extensive research conducted on the use of neural networks as a tool for prediction an optimization, no ready-made solution to an ANN model can be given. In AI-based solutions, a so-called tailored made solution has to be developed for every dilemma. This phenomenon occurs due to the fact that every company has specific activities and therefore specific data to work with. Furthermore, not a lot of literature is available on the use of neural networks for creating a cost estimation method to estimate the cost of engineering services[1]. Considering the existing problem within the company, the emerge of AI and the cohesive proposed AI estimation methods, the research focuses on the development and evaluation of an artificial neural network solution to estimating the costs of services of engineering consultancy firms.

[1] To the best of knowledge there is only one study that contributes towards using neural networks for estimating cost of engineering services. This is the study proposed by Hyari et al. (2016). However, they focus on a different market namely public construction. In addition, they try to predict engineering services costs as a percentage of the total construction costs based on quantitative data. In this study also qualitative data is used and a price in euro's is modelled. Also, they only use 5 variables and do not show how the relative importance of thet these variable is.

2.6 Machine learning methodology

Now that it is known that a modern AI estimation method that can potentially overcome the research problem and can reach the research objective, this method is being explained more in-depth. The machine learning methodology consists of several specific elements and the right elements for this research needs to be distinguished. Machine learning is a field in computer science where existing data are used to predict, or respond to, future data (Paluszek & Thomas, 2017). The methodology is closely related to the fields of pattern recognition, computational statistics, and artificial intelligence. In areas like facial recognition and spam filtering, it is not feasible or even possible to write an algorithm to perform the intended task that is where machine learning gets important. Machine learning is a technique that figures out the "model" out of "data" (P. Kim, 2017). It does so by learning from training data, without being explicitly programmed. In Figure 2-2 below the machine learning process is illustrated and shows what happens in the machine learning process. First of all, a training dataset is exposed to the machine learning technique and after the learning process, a model is established. This model is the end product of the machine learning methodology and can be used for implementation in practice. The model can then be provided with new input data that is not known to the model and can give an output based on the patterns or relationships identified in the training data.

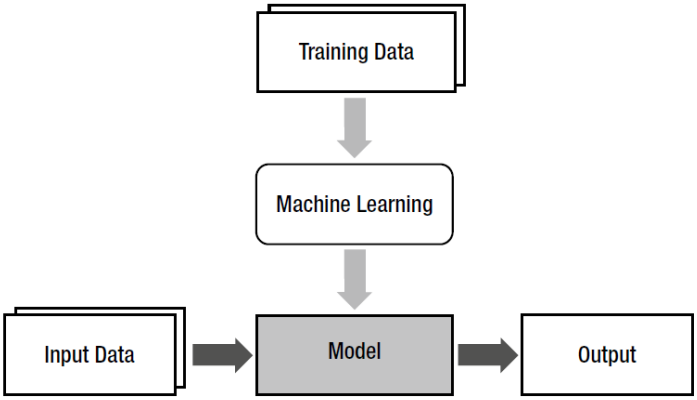


Figure 2-2. Machine learning process

2.7 Elements of machine learning

Machine learning methods are data driven. Datasets are usually collected by humans and used for training (Paluszek & Thomas, 2017). Just as humans need to be trained to perform tasks, machine learning systems also need to be trained. There are different types of machine learning techniques, these can be classified into three types depending on the training method. The three different techniques are: (1) supervised learning, (2) unsupervised learning, and (3) reinforcement learning. The three different techniques are illustrated in Figure 2-3 below.

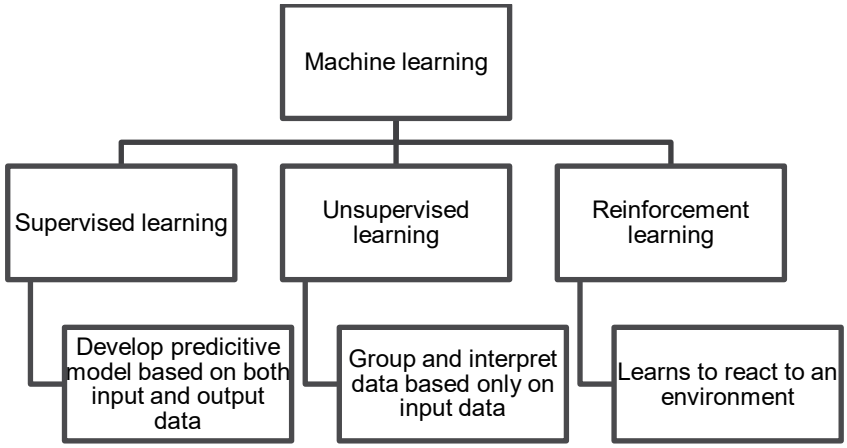


Figure 2-3. Different types of machine learning

2.7.1 Supervised learning

In a supervised machine learning technique, the learning process is based on datasets that provide both input values as output values. The process is called supervised as the patterns in the data are recognized using the correct corresponding output values of the input values. The supervised learning technique is similar to the way humans learn. Humans apply current knowledge to solve a problem, then comparing the answer with the solution. If the answer is wrong, the current knowledge is modified in order to solve the problem better the next time. In supervised learning, this is done by the series of revisions of a model to reduce the difference between the correct output and the output of the model for the same input. The important aspect of supervised learning is that the solutions are needed to make it work.

2.7.2 Unsupervised learning

In unsupervised learning, the learning process is only based on the input values and is used for situations where no “right” answer is known. This technique identifies patterns in data and reacts based on the presence or absence of such commonalities in each new piece of data. Clustering algorithms are generally examples of unsupervised learning. The biggest advantage of unsupervised learning is that it can learn aspects of data that are not known in advance, finding hidden structures in data.

2.7.3 Reinforcement learning

Reinforcement learning uses sets of input, some output, and grade as training data. It is generally used when optimal interaction is required, such as control and gameplays (P. Kim, 2017). It is mostly concerned with dealing with problems in an active environment in which the specific situation requires specific actions to take. The focus is on performance, which involves finding a balance between exploration of uncharted territory, and exploitation of current knowledge (Busoniu, L., Babuska, R., De Schutter, B., & Ernst, 2010)

2.8 Application of machine learning

In Figure 2-4 on the next page, the different machine learning techniques and examples of their application are illustrated. The objective of this research concerns the development of a method for estimating the cost of engineering services in the tender phase. This is a problem where the solution of the data is known, these solutions are known due to the availability of project information about past projects. Therefore a supervised machine learning approach is used. For supervised learning there exists classification problems and regression problems. If the output variable is continuous in nature than there is a regression problem. Whenever the output variable describes some categories or discrete classes than there it concerns a classification problem. The objective in this specific research is concerned with predicting a continuous value and thus this research is concerned with a regression approach. Regression predictive modelling is the task of approximating a mapping function (f) from input variables (x) to a continuous output variable (y). In Figure 2-5 an example of a nonlinear regression fit is shown. One of the most commonly used algorithms in regression analysis for cost prediction of construction projects is an artificial neural network. Since neural networks can have many layers (and thus parameters) and are capable to work with non-linearity, they are very effective at modelling highly complex non-linear relationships. Therefore, neural networks are used in achieving the research objective.

Figure 2-4. Application of different machine learning techniques

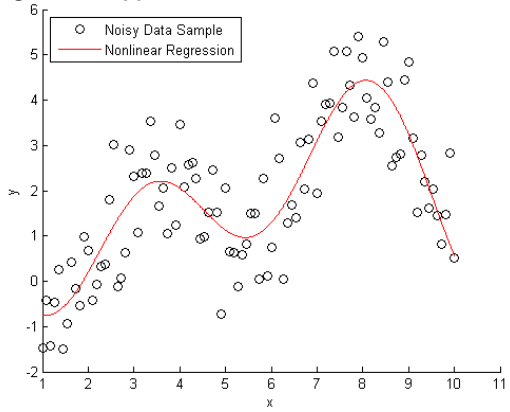


Figure 2-5. Example of a nonlinear regression fit

2.9 Artificial neural networks

ANNs are originally inspired by the study of processes in the human brain (Günaydin & Doğan, 2004). The human brain acquires knowledge through a learning process, whenever we learn something, our brain stores the knowledge. This principle is the same with ANN and the inter-neuron connection strength known as synaptic weights are used to store the knowledge. ANNs consist of nodes (neurons in ANNs) grouped in interconnecting layers and sets of layers to form a network (Petroutsatou et al., 2012). There are three different types of layers namely; input, hidden and output layers. The layout or architecture of a network can be viewed below in Figure 2-6. In this chapter, the principles behind ANNs are described by the use of the book that is published by P. Kim (2017).

Initially, neural networks had a very simple structure with only input and output layers, these were called single layer neural network or shallow neural networks. Neural networks with multiple hidden layers are called multi-layer neural networks or deep neural networks. A neural network with this structure can also be referred to as a multilayer perceptron. Most of the contemporary neural networks used in practical applications are deep neural networks (Kim, 2017). Every input node has a connection with all the nodes from the next hidden layer. This connection is illustrated by the arrow in the figure below and is corresponding with a particular weight.

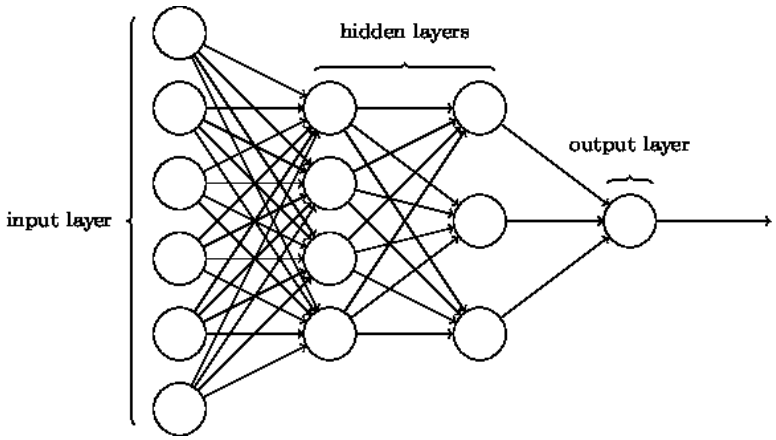


Figure 2-6. Structure of deep neural network or multilayer perceptron

Training a neural network

In Figure 2-7 below the supervised learning process of a neural network is illustrated. Before the training process can begin, a dataset with input variables and the corresponding output variable is needed. Every pair of input variables and the corresponding output variable is called a data point. The network is trained with the use of the established training data from the dataset. The learning process consists of three different steps which are carried out in an iterative process. The three steps are as follows:

1. Feedforward propagation: Take the input and correct output from the training data and enter it into the neural network. Obtain the output from the neural network and calculate the error with the correct output.
2. Backpropagation: Calculate the error contribution in each node, and adjust the weights accordingly to reduce the error.
3. Repeat Steps 2-3 for all training data.

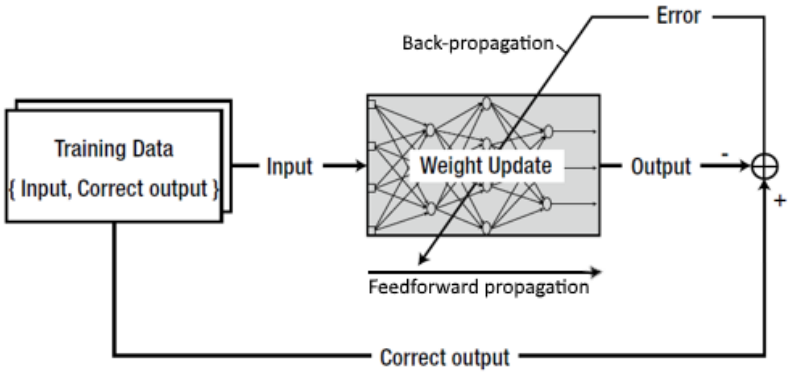


Figure 2-7. Supervised learning concept

Feedforward propagation

First of all, the weights of a network need to be initialized. When first training a network the weight initialization is done randomly. For multilayer networks, the weights and biases are generally set to small random values ranging between -0,5 and 0,5. The training of a network begins with a principle called feedforward propagation. The input layers receive the inputs from a data point and direct them to the hidden layers, without calculations. The nodes in the hidden layers and output layers perform the computations of the network and add and adjust weights. The weight is expressed in a numerical value. Every hidden node takes the weighted input of the previous node and outputs a single value based on a predefined transfer function (Bosscha, 2016). Eventually, the neural network provides an output based on the input and configuration of the weights. In Figure 2-8 below a better understanding of the neural network's mechanism is explained using a node that receives thee inputs.

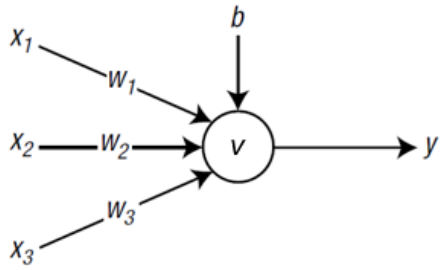


Figure 2-8. A node that receives three inputs

The circle and arrow in the figure above illustrate the node and signal flow, respectively x_1, x_2, and x_3 are the input signals. w_1, w_2, and w_3 are the weights for the corresponding signals. Lastly, b is the bias, which is another factor associated with the storage of information. In other words, the information of the neural network is stored in the form of weights and biases (Kim, 2017). The input signals or input variables, in a vector ($\vec{x}$) are multiplied by the weight vector ($\vec{w}$) before it reaches the node. The weighted signals are collected and summed up to be the weighted sum, lastly the bias, (b) is summed up and the total weighted sum (v) is calculated (Equation 1). The equation can also be written by using vectors (Equation 2).

$$v = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3 + b \quad \text{(Equation 1)}$$

$$= \vec{w} \cdot \vec{x} + b \quad \text{(Equation 2)}$$

Subsequently, the node enters the weighted sum (v) into a transfer function $\varphi(v)$ and yields its output y (Equation 3). The transfer function determines the behaviour of the node. In most cases, tan-sigmoid transfer functions (Equation 4) are used in the hidden layers and linear transfer functions in the output layer (Janssen, 2018). However, this depends on the problem that the network is trying to solve. The output is passed outside to other nodes in the next layer. Eventually, the output node in the output layer gives the output of the network.

$$y = \varphi(v) = \varphi(\vec{w} \cdot \vec{x} + b) \quad \text{(Equation 3)}$$

$$\varphi(v) = \frac{e^v - e^{-v}}{e^v + e^{-v}} \quad \text{(Equation 4)}$$

Whenever a neural network has been established, the configuration of the weights initially leads to an error between the output of the neural network and to the correct output. Therefore, at this point, the neural network is useless in any application. This means that the configuration of the weights need to be optimized in a way that the error between the output of the network and the correct value is minimized. This is done by using a form of back-propagation.

Back-propagation

The process of adjusting the weights of the neural network is based on the calculated error and is carried out using a so-called back-propagation algorithm, the representative learning rule of the multi-layer neural network. An example using a simple neural network is used to explain the principles behind the back-propagation algorithm. In the back-propagation algorithm, the delta of the output node (δ_n) needs to be calculated by the application of the generalized delta rule[2] (Equation 5). The generalized delta rule is used to calculate the delta (δ_n) of a hidden or output node. To do this, the derivative of the transfer function of the output node is used and the weighted sum of the corresponding input node is put into the function $\varphi'(v_n)$. The derivate function calculates the slope of the non-derivative function. Whenever the output of the derivative function of the weighted sum is relatively high, the slope is steep, which means that the output of the node is far from reaching 1 or -1 (see Figure 2-9). The higher the output of the derivative function, the more the node is contributing to the total error, and the more it needs adjustment.

[2] In machine learning, the Delta rule is a gradient descent learning rule for updating the weights of the inputs to artificial neurons in a single-layer neural network. A generalized form of the delta rule, developed by D.E. Rumelhart, G.E. Hinton, and R.J. Williams, is needed for networks with more than one hidden layers.

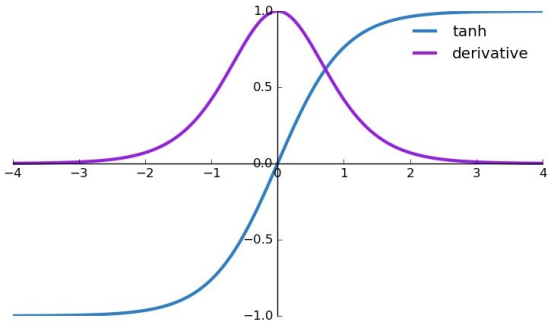


Figure 2-9. Sigmoid function and derivative sigmoid function

Subsequently, the output of the derivative function needs to be multiplied with the error of the output node. In the generalized delta rule, e_n is the error of the output node which is calculated by subtracting d_n, which is the correct output, from y_n, which is the output of the network (Equation 6). In Figure 2-10 below an example of a simple neural network is provided to illustrate the deltas of the output nodes.

$$\delta_n = \varphi'(v_n) \cdot e_n \quad \text{(Equation 5)}$$

$$e_n = d_n - y_n \quad \text{(Equation 6)}$$

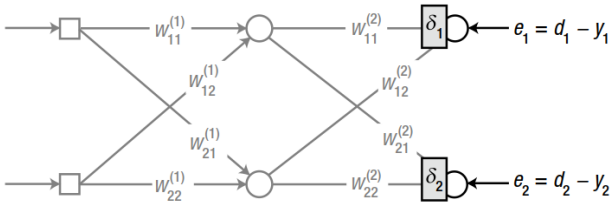


Figure 2-10. Back-propagation training algorithm

Now that the delta of the output node is known, the delta's in the hidden layers need to be calculated. For a visualization of this process see Figure 2-11. To do this first the error of the hidden node needs to be calculated. This is done by taking the weighted sum of the back-propagated deltas from the layer on the immediate right (Equation 7). In the equation, an example is provided to calculate the error $e_1^{(1)}$. Once this is done the calculation of the delta of the hidden node $\delta_1^{(1)}$ can be calculated using the generalized delta rule (Equation 8). For every hidden node, the error and the delta can be calculated and the delta of the hidden node is passed outside to the immediate left.

$$e_1^{(1)} = w_{11}^{(2)} \cdot \delta_1 + w_{21}^{(2)} \cdot \delta_2 \quad \text{(Equation 7)}$$

$$\delta_1^{(1)} = \varphi'\left(v_1^{(1)}\right) \cdot e_1^{(1)} \quad \text{(Equation 8)}$$

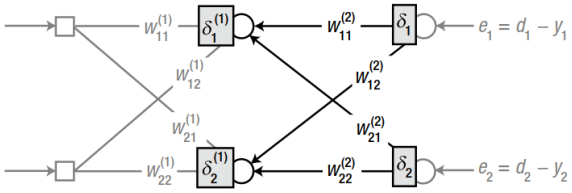


Figure 2-11. Leftward proceeding calculating delta in hidden nodes

The last step is to update the weights, this is done by an example of updating weight $W_{21}^{(2)}$. The formula (Equation 9) below is applicable for all the different weights in the network, as all the delta's within every node can be calculated. In this equation $W_{21}^{(2)}$ is the highlighted weight in Figure 2-12 below, α is the learning rate of the network, δ_2 is the delta of the node, $y_1^{(1)}$ is the output of the first hidden node. The learning rate is the rate in which the network learns, this can be altered depending on the problem at hand. Weight update is carried out by using the fundamental concept where the weight is determined in proportion to the output node error, and the input node value.

$$W_{21}^{(2)} = W_{21}^{(2)} + \alpha \cdot \delta_2 \cdot y_1^{(1)} \quad \text{(Equation 9)}$$

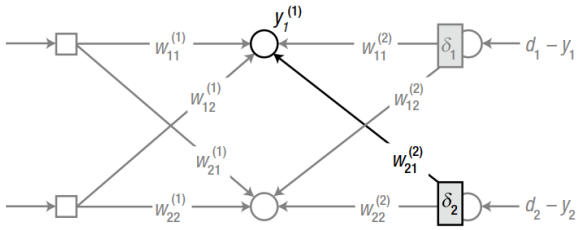


Figure 2-12. Adjusting weights

Repeat steps

The process of feedforward propagation and back propagation is carried out for all the data points that are present in the training data. After that, the whole process can be repeated with the same training data. Every training iteration in which all the data points in the training data are used once is called an epoch. For example, if an epoch of 10 is used, every data point is used 10 times in optimizing the network. In addition, the example explained above concerns the basic form of backpropagation. However, the basic form of back-propagation is too slow for most practical applications. In machine learning, there are a lot of variations of backpropagation that provide significant speedup and make the algorithm more practical. These variations of the backpropagation algorithms are all derived from the basic backpropagation algorithm. In the next paragraphs, the principles behind making a profound choice for training algorithm, network architecture, transfer function, and network performance function are explained.

2.9.1 Selecting a training algorithm

There are several variations of the basic backpropagation, which are faster and more practical than the basic backpropagation. These can be trained and updated either in batch mode or sequential mode. In batch mode, the weights within the network are updated after all the input are presented to the network. This update is based on the total gradient that is determined by summing up the gradients for each input. For the sequential mode, the weights are updated after each input is presented to the network. Usually, for many of the more efficient optimization algorithms, the batch mode is inherent. For a relatively small network that has up to a few hundred weights and biases that are activated for function approximation, the Levenberg-Marquardt algorithm is the fastest training method. In addition, the study performed by Hyari, et al. (2016) showed that the best performing training algorithm was the Resilient Backpropagation algorithm. Due to the similarities that study has with this research, the resilient back-propagation algorithm is also tested. Lastly, in order to see what the effect of regularization is for improving generalization, the Bayesian regulation backpropagation algorithm is also tested. The following algorithms selected and are tested in this research:

Levenberg-Marquardt backpropagation

The Levenberg-Marquardt is often the fastest backpropagation algorithm and is highly recommended as a first-choice supervised algorithm. However, it does require more memory than most of the other algorithms. The Levenberg-Marquardt algorithm is used to solve non-linear least squares problems. The best fit in the least-squares sense minimizes the sum of squares between an overserved value and the fitted value provided by the model. This algorithm makes use of early stopping as a way to improve generalization. It does so by means of a validation set. Whenever the performance of the validation sets becomes worse for a set amount of iterations, the training stops. The networks trained with this function must use either the Mean Squared Error (MSE) or Sum Squared Error (SSE) performance function.

Bayesian Regularization backpropagation

The Bayesian Regularization algorithm is a network training algorithm that updates the weights and bias values according to the Levenberg-Marquardt optimization. It minimizes a combination of squared errors and weights and determines the best combination so as to produce a network that generalizes well. In contrary to the Levenberg-Marquardt algorithm, this algorithm makes use of regularization as a method to improve generalization. As this training algorithm makes use of the Levenberg-Marquardt optimization, it also must use either the Mean Squared Error (MSE) or Sum Squared Error (SSE) performance function.

Resilient backpropagation
This algorithm was created by Riedmiller & Braun (1992). Multilayer network typically uses sigmoid transfer functions in the hidden layers. Sigmoid functions are characterized by the fact that their slope must approach zero as the input for that function gets large. This can have problems when using the basic backpropagation algorithm, as relatively large input mistakes can result in a small magnitude of the gradient. Therefore, this can cause small changes in the weights and biases. This problem is solved by the resilient backpropagation algorithm to update weights by an alternative update value (Ki & Uncuo, 2005). This algorithm can, in contrary to the other algorithms, make use of Mean Absolute Error (MAE) as a performance function.

2.9.2 Select network type and architecture

The selection of the network type concerns choosing the correct transfer function and type of problem. The network architecture concerns choosing the correct number of layers, hidden neurons, and input variables. After the training algorithm is selected, the network type needs to be determined and is defined by the problem that is solved. Hagan et al., (2014) identifies four types of problems namely: pattern recognition, clustering, prediction, and fitting. In pattern recognition, a neural network tries to classify the input into a set of target categories. For example, a physician might want to classify a tumour as benign or malignant based on uniformity of cell size, clump thickness and mitosis. For clustering, the neural network tries to group data by similarity. For example, businesses can perform market segmentation, which is done by grouping people according to their buying patterns. Prediction concerns the prediction of a future value of some time series, this is used in for example in stock trading. Pattern recognition, clustering, and prediction do not cover the problem of this research. In chapter 2.8 it was already determined that the problem of this research concerns a regression problem. Fitting is also referred to as function approximation or regression and therefore this research concerns a fitting problem.

The most commonly used and standard neural network architecture for fitting problems is the multilayer perceptron[3]. In most cases, tan-sigmoid transfer functions are used in the hidden layers and linear transfer functions in the output layer (Janssen, 2018). The tan-sigmoid transfer function provides normalized output values for the hidden node between -1 and 1, which is similar to the normalized input data. Due to this similarity, the data is less saturated and preferred above the log-sigmoid function that provides an output range for the hidden nodes between 0 and 1.

Most of the fitting problems perform sufficient with the use of a single hidden layer. However, in some cases, two hidden layers are used when performance is lacking. It would be very rare in standard fitting problems to use more than two hidden layers. Therefore the training will be started with one hidden layer, and subsequently, a training session with two hidden layers will be tested. Based on the performance either an architecture consisting of one or two hidden layers is chosen. Thereafter, the number of neurons in each layer needs to be determined. The number of neurons in the output layer is the same as the size of the target vector or output vector and is determined by the goal it tries to achieve.

The number of neurons in the hidden layers and input layers, however, can have an influence on the ability of generalization a neural network has. Generalization is the concept of getting only the wisdom from the data that is in it. A network trained to generalize well will perform as well in new situations as it does on data on which it was trained. The complexity of a neural network is determined by the number of free parameters it has (weights and biases). The number of free parameters is in their turn determined by the number of neurons. If a network is too complex for the amount of data that it is trained with, it will most likely overfit and generalize poorly. If a network is too simple, it will most likely underfit and also generalize poorly. The concepts of underfitting and overfitting are also illustrated in Figure 2-13 below.

[3] The layout of the multilayer perceptron is illustrated in Figure 2-6 of this report.

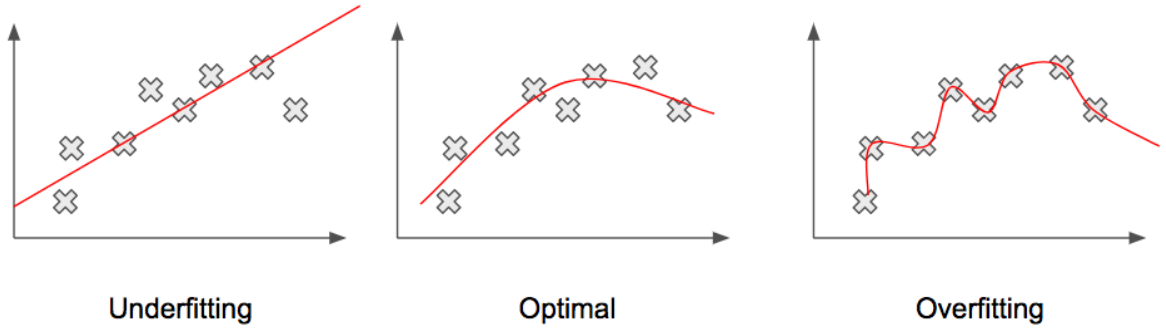


Figure 2-13. Concepts of underfitting and overfitting

The number of neurons in the input layer is the same as the number of variables that are used in the database. However, sometimes input vectors have redundant or irrelevant elements. Especially when the input vector is large, it can be beneficial to eliminate redundant or irrelevant elements. This can assist in preventing overfitting during training, reduce required computation effort and enhance generalization. A method to determine the absolute importance of each input does not exist. However, a sensitivity analysis can be helpful to determine relative importance. Therefore a sensitivity analysis will be carried out to determine and eliminate potential redundant variables. Olden & Jackson (2002) propose a method to determine the relative importance of input variables on the predicted output. This relative importance is calculated using the magnitude of the weight per independent variable (Ibrahim, 2013; Janssen, 2018).

Lastly, the number of neurons in the hidden layers are determined by the complexities of the function that is being approximated. It is not known how complex the problem is until the network is trained and analysed. The best number of neurons in the hidden layers is therefore determined empirically. By adjusting the architecture of the network, the performance of the model can improve. The key to creating a network that is able to generalize well is to find the simplest model that explains the data. In order to do so, a network is build that contains the smallest number of free parameters that still explains or fits the data well enough. Hagan et al., (2014) explains that there are five different approaches that are generally used to produce simple networks: growing, pruning, global searches, regularization and early stopping. In this research, the growing technique is used. Growing means starting with zero neurons in the hidden layer and increase the number of neurons until the desired performance is achieved. To achieve this, a strategy towards the determination of the architecture of the network is established. The strategy towards creating a network that is able to generalize well is explained in the proposed method that is explained in chapter 3.

2.9.3 Initialize weights and train network

Before the training of the network can start, the weights and biases need to be initialized. The approach for initializing the weights and biases depends on the type of network. For multilayer networks, the weights and biases are generally set to small random values ranging between -0,5 and 0,5, if the inputs are normalized to fall between -1 and 1. It could happen that a single training run may not produce optimal performances (Janssen, 2018). This could happen because of the possibility of reaching a local minimum of the performance surface. In Figure 2-14 below the difference between a local minimum and global minimum is illustrated.

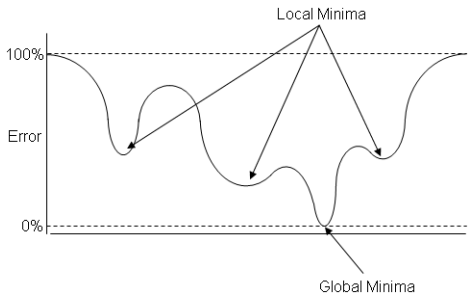


Figure 2-14. Local minimum vs global minimum

It is best to restart the training process using several different initial conditions. The training of a multilayer neural network is involved with two stochastic elements, due to which every training run a different performance will emerge. The first stochastic element regards the initialization of weights and, biases, which is done randomly every training run. The second stochastic element is the random division of training, testing and validation set (Beale, M. H., Hagan, M. T., & Demuth, 2018). Due to the fact that in every training run different initial conditions are set, restarting the training process several times is advised to find the best performance. Therefore, a so-called 'multistart' is used to restart the training process 100 times.

2.9.4 Analyse network performance

When training a neural network, the performance should be analysed in order to determine if the network training was successful. In order to analyse the performance first a choice of a performance function needs to be made. For multilayer networks, the standard performance index is mean square error (MSE). The equation for the MSE can be defined as described in Equation 10 below. Where N is the number of data points, f_i the value returned by the model and y_i the actual value for data point i.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2 \qquad \text{(Equation 10)}$$

In addition, one useful tool for analysing neural networking that tries to solve fitting problems is regression analysis. This is done by determining and calculating the regression between the trained network outputs and the corresponding targets. The targets could be plotted on the x-axis of a graph and the network output on the y-axis. For a perfect fit of the model, the data should fall along a 45-degree line, where the network outputs are equal to the targets. When data points are deviating relatively far from the regression line, these data points can be characterized as outliers. These outliers can then be investigated further, it could mean that an outlier concerns a bad data point or that the point is located far from the rest of the training data. In the latter case, this would mean that more data is needed in that region.

Based on the results a correlation coefficient can be computed, which is also known as the *R*-value. The *R*-value can range between -1 and 1, however, for neural network application it is expected to be close to 1. Whenever *R* = 1 the data is fitting perfectly, and when *R* = 0 the data will be randomly scattered and not have a fit at all. When *R* values are significantly lower than 1, the neural network is not properly fitted to the underlying function. Then effort should be made towards determining if there are bigger outliers in different regions of the data. For example, targets with higher values could have more outliers and this could mean that more training data is needed for target values in that range. The *R* value needs to be similar in all sets (training, validation, and test), in order to ensure good generalization. Whenever significant differences between sets arise, overfitting or extrapolation could be the case. If the validation error or test error is much larger than the training error, this could mean that overfitting has occurred. If all three sets have larger errors, this could mean that the network is not powerful enough to fit the data. Lastly, an error histogram is a powerful tool in order to see how many errors fall within a particular interval of errors. For example, how many data points fall in the interval of an error of 10.000.

The training phase will be carried out with the evaluation of the performance index MSE, however, the eventual network is chosen based on the performance index mean absolute percentage error (MAPE). This metric calculates the average of the absolute values of the differences between each predicted output and the corresponding target output, as shown in equation 11. Where N is the number of data points, F_t the value returned by the model and A_t the actual value for data point i.

$$MAPE = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{A_t - F_t}{A_t}\right| \qquad \text{(Equation 11)}$$

Regression model fitting minimizes the squares of the errors, so models developed using this technique will be inherently biased towards minimizing the errors for large projects where errors are greatest. What is important about the MAPE metric is that it penalizes huge errors not as badly as MSE does. This due to the fact that MAPE is a linear score which means that all the individual differences are weighted equally in the average. The MSE is quadratic and takes bigger error more into account then smaller errors. Therefore MSE is scale depended and MAPE is not. In addition, the MSE performance measure is more sensitive towards outliers and can detect them quicker than while using MAPE. The performance metric selected should be based upon whether the user wants accuracy in proportional or scaled terms (Emsley, Lowe, Duff, & Hickson, 2002). In this research, the MAPE performance metric is used for selecting the best model when the training of the model is finished. However, the

model is only selected if the R values of the training set and test set are relatively similar and close to 1. When this is not the case, the next best MAPE is chosen.

2.9.5 Data comparison earlier work

The quality and amount of training data is often the single most dominant factor that determines the performance of a model. The amount of data that is needed for a machine learning algorithm depends on the complexity of the problem and on the complexity of the chosen algorithm. A significant amount of practitioners have worked on a lot of applied machine learning problems before. Therefore, reasoning by analogy is a way to determine the amount of data that is probably needed. Basically, this means to investigate similar machine learning projects and see the amount of data is used in these studies and evaluate the performance. In Table 2-5 below, 8 different studies that are somewhat similar to this study are provided. For these studies, the number of datasets that are used in the development of the machine learning algorithm can be viewed and the coherent performance of the model. All the performances are given in the mean absolute percentage error (MAPE). It can be seen that when fewer data points are used in a model this not immediately corresponds with lower performance. For example, Hyari et al. (2016) used 224 datasets and achieved a performance of 28,2% which is lower than the 10,4% that Cheng et al. (2010) achieved with only 28 data points. This shows that although they try to solve relatively similar cost estimation problems, the amount of data does not have a very direct relationship with the performance. Due to the fact that the complexity of the problem depends on a lot of things, the performance of a model is usually determined empirically. Therefore, the amount of data that is needed for good performance is not known before training the model. The results of the analysis of the performance of the network can help to decide if we have enough data (Hagan et al., 2014). Therefore in this research, as much data as possible is collected in the time that is available.

Table 2-5. Comparison with earlier work

No. of data points	Performance	Sources:
28	10,4% (MAPE)	(Cheng et al., 2010)
30	7% (MAPE)	(Günaydin & Doğan, 2004)
224	28,2% (MAPE)	(Hyari et al., 2016)
288	16,6% (MAPE)	(Emsley et al., 2002)
71	4% (MAPE)	(Mohammed Arafa and Mamoun Alqedra, 2011)
52	17% (MAPE)	(Mahamid, 2013)
813	6,2% (MAPE)	(Arage & Dharwadkar, 2017)

2.10 Proposal price influencing factors

Now that is known what is important while developing an ANN cost estimation method, the proposal value influencing factors can be distinguished. In the supervised learning process of an ANN, the learning process is based on datasets that provide both input values as output values. This is done by identifying the patterns in the data are recognized using the correct corresponding output values of the input values. Therefore, these input values or factors that influence the output value need to be identified. In this research, the proposal price is used as the determined output value for the ANN. The model should predict the proposal price based on a set of input values. An initial effort to identifying factors that influence the proposal price is done by evaluating the literature.

Akintoye (2000) conducted a factor analysis and principal component analysis based on 24 different factors that influence project cost (Capex) estimating in the construction industry. The cost factors that are proposed by Akintoye (2000) are not all relevant for the cost estimation of engineering services. Other studies provide information about the relevant factors for engineering services. Zwaving (2014) did research that contributed to a probabilistic estimating approach for cost estimation of engineering services within the energy and chemical industry. In their research, they propose a set of factors that are relevant for cost estimation of engineering services. Furthermore, Hyari, Al-Daraiseh, & El-Mashaleh (2016) did research concerning the development of a conceptual cost estimation model for engineering services in a public construction project. Based on Akintoye, (2000), Hyari et al. (2016) and Zwaving, (2014) a table with the relevant cost factors for estimating the costs of engineering services is established (Table 2-6).

Table 2-6. Cost factors that affect project cost estimating for engineering services

Number	Cost factor
1	Scale of work
2	Scope of work
3	Project duration
4	Quality of information and information flow
5	Pre-contract design (extent of completion of pre-design)
6	Type of client and requirements
7	Complexity of design and construction
8	Project team experience
9	Number of project team members
10	Type of work
11	Project phases
12	Design changes
13	Collaborating disciplines
14	Market conditions

Scale of work
The scale of the work is usually expressed in capital expenditure or Capex. This is the total investment of the construction in euro. The larger the project it is more likely that there is more work to be done for the engineering firm. Therefore, the scale of work expressed as the Capex can have an influence on the costs of engineering services.

Scope of work
In a project, an engineering consultancy firm can take different roles and can perform different activities and services. These services are usually procured by the client in three main packages namely: Engineering (E), Engineering, Procurement and Construction (EPC) or Engineering, Procurement and Construction Management (EPCm). Engineering concerns only delivering the design package. EPC concerns delivering the design, procurement of subcontractors and material and the management of the construction on behalf of the client. EPCm concerns the same as EPC only the financial risk lays with the contractor or engineering firm.

Project duration
The expected project duration can have an influence on the costs of engineering services. For example, when the time that is needed to carry out a project has not carefully planned ahead, extra costs can emerge due to the fact that the product is not delivered on time. In addition, construction may step in while the engineering has only partly been performed this often causes some rework to be done (Zwaving, 2014). The project duration is usually expressed in weeks or months.

Quality of information
It is often emphasized in the literature, that the quality of information about a project is an important factor that affects the accuracy of a final estimate (Lester, 2017). Often, there are more risks and uncertainties during a project when the quality of information lacks. This could eventually lead to higher costs during the project due to these unforeseen circumstances. The most important factor causing estimation error in constant dollar terms appears to be the level of process and project definition when an estimate is made (Merrow, S.W., & Worthing, 1979).

Pre-contract design
The extent of completion of the pre-design determines how much work still has to be performed in order to deliver the required documents or results at the end of a project. The quality and extent of completion of a design could differ in the entry of a project. Therefore, when the extent and quality of a previous design lack, more work needs to be done in the next phase.

Type of client and requirements
The type of client is also important for the total costs of engineering services. Every client has different demands of delivering the design and services. For example, a client could have strict guidelines on documentation that is needed along with the design. This could lead to additional hours of work for the engineering consultancy firm. On the contrary, there are also clients who only want the final design in the form of a 3D model and 2D drawings. This requires fewer hours to deliver the final result and still meet client demands.

Complexity of design and construction
The complexity of a design determines the required amount of work that is needed to complete the work. The complexity of a project is hard to determine in advance, however from analysing the scope definition it can become clear were complexities or risks lay within a project. Furthermore, more complex projects require more experienced and more expensive engineers or project managers. In addition, more complex projects require more time and effort.

Project team experience
When a project team has experience with a certain project, it can finish the project faster and more efficient due to the learning curve they already have experienced. Whenever the project team is less experienced with a certain project, they still need to go through a learning curve and eventually the project will most likely take longer to complete. In addition, the project team experience can also increase the quality of a design and therefore reduce failure costs.

Number of project team members
The number of project team members that are required in a project has an expected correlation with the total amount of work that is required. The more team members the project requires, the more expensive the project becomes. Furthermore, the number of project team members determines the amount of collaboration that is needed in a project to align all the work. Therefore, when there are more project team members the total amount of work or hours could be higher.

Type of work
The type of work regards to whether a project concerns a new construction (greenfield project) or maintenance or expansion of an existing construction (brownfield project). When the project concerns an expansion of an existing construction the dimensions and characteristics of the design need to be identified and known. This will lead to more work to identify and get insight into this existing design. In general, existing construction leads to more work as there are more restrictions and constraints within a project.

Project phases
In total there are three different phases of designing before the start of construction namely: Conceptual development, preliminary engineering, and detailed engineering. For these phases, different percentages of the total design need to be carried out. For example, more work needs to be done for a detailed design than a conceptual design and is, therefore, more expensive. In addition, for a different level of design, the engineers usually have a different level of experience. Therefore, the project phase is important for the determination of the costs of engineering services.

Client's attitude towards design changes
The client's attitude towards design changes can be described as the level of cooperation towards approving design change notices (DCNs). A client can be very cooperative or could be uncooperative. The client's attitude towards design changes could increase costs for the engineering consultancy firm. First of all the additional work that is required due to the change in the scope of the design increases costs. When clients are difficult with regard to approving these DCNs, higher costs can arise within the project.

Collaborating disciplines
This variable concerns the number of different disciplines that are collaborating in a project. In larger projects, multiple offices or disciplines can work together at the same time. Whenever there are multiple disciplines or offices active in a single project, additional coordination is needed to align the work of the different disciplines and offices. This could lead to more work in the domain of project management and will increase the costs of engineering services. In addition, larger projects usually require work that needs to be carried out in more and different fields of expertise, which is inherent to the disciplines. Therefore, it also implies to the size of a project.

Market conditions
This variable is about the different market conditions that the project can experience. Engineering consultancy firms usually make designs and deliver work within different markets (e.g. Oil & Gas, Food, Pharma). Within these industries, different standards are valid and different products need to be produced. For example, different types of drawings need to be produced and delivered, or more detailed specifications need to be provided with the produced drawings. In Food more effort need to be made towards hygiene, and for Oil and Gas more effort need to be made towards safety. Therefore this could potentially affect the costs of engineering services.

3 PROPOSED METHOD

In this section, the proposed method that is used for model development is described. The proposed method covers phases 3, 4 and 5 of the research strategy. The proposed method (see Figure 3-1 below) consists of the development and training of the ANN and is done with the use of the book published by Hagan, Howard, Demuth, & Beale (2014). Hence, the model development process can be subdivided in three main aspects, namely: pre-training steps, network training and post-training analysis (Hagan et al., 2014). These three main aspects correspond respectively to the phases 3, 4 and 5 of the research strategy.

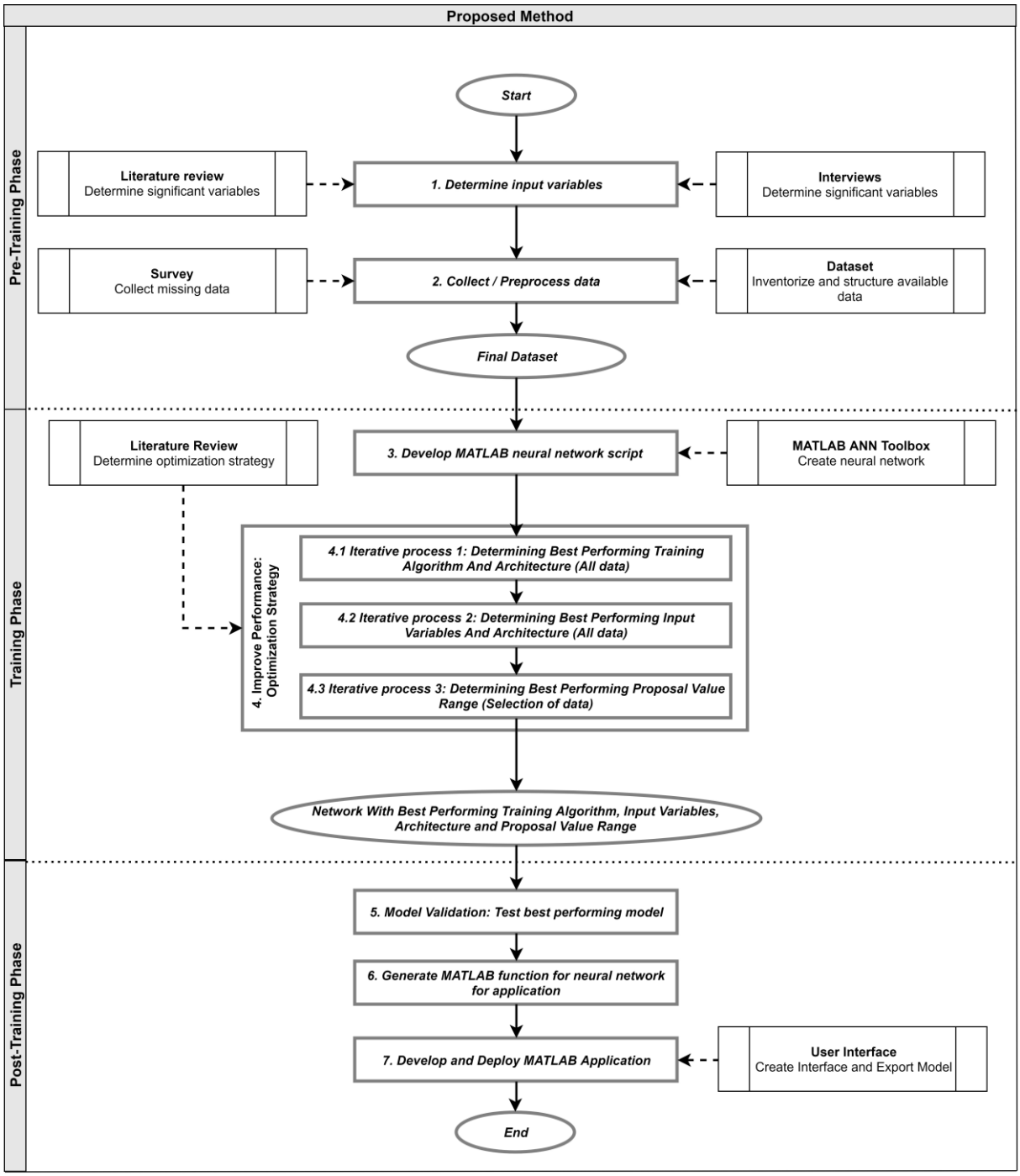


Figure 3-1. Proposed method

The pre-training phase concerns the establishment of the dataset and in this phase the input variables of the model are determined. Furthermore, the data that is needed to train the model is collected and pre-processed. Subsequently, the training phase covers the development of the actual model. The training phase consists of creating an ANN in MATLAB and improving the performance of the model by carrying out an optimization strategy. The optimization strategy consists of three iterative processes. The first iterative process is about determining the best training algorithm and best network architecture, this is done using the full dataset. The second iterative process will determine whether the model can have better performance by using fewer input variables. The third iterative process in the optimization strategy consists of testing different proposal value ranges. Lastly, the post-training phase is about building an application and internally validating the model. The proposed method will be described in this chapter.

3.1 Pre-training phase

The pre-training phase concern the selection of data and data pre-processing. Neural networks represent a technology that is at the mercy of the data. The training data must span the full range of the input space for which the network will be used (Hagan et al., 2014). Neural networks can interpolate accurately throughout the range of the data preceded, however extrapolation outside the range of the training set is of lower quality. To ensure the right data is used in training the model, a selection of data is made. First of all, the input variables that influence the proposal price that are identified in the literature review are further developed by interviews. Based on the availability of the input variables, a survey is established and spread to gather potential supplementary data. Lastly, the data should be pre-processed for an efficient training process.

3.1.1 Determine input variables

To start, the factors that influence the proposal price that are determined in the literature review need to be further investigated and assessed. The literature review led to the identification of the possible variables that influence the cost of engineering services (see chapter 2.10). This process is done by carrying out desk research and reviewing different journals, papers, and essays. In order to verify and determine additional influencing factors, interviews are carried out with experts within the research client. Interviews were held with 13 employees within Bilfinger Tebodin that have experience with preparing bidding offers for engineering services. The interviewees consisted of three project managers, five lead engineers (different departments), two heads of departments and three tender managers. They were asked to answer 6 open questions that lead to the most important factors that influence the costs of engineering services. The following 6 questions were asked:

1. Can you describe the approach you would use in order to estimate the required man-hours of a project based on a Request For Quotation (RFQ)?
2. Can you describe the approach you would use in order to estimate the required man-hours of a project based on a RFQ if you only had one hour?
3. Can you describe how you get a broad view of the size of a project while reviewing a RFQ?
4. What information or elements are ideally available in a RFQ to make an estimate?
5. If you had to make a proposal without an RFQ and you could ask 5 questions to the client, what questions would you ask?
6. Can you explain what the variables are that influence the costs of engineering services in a project?

While answering these question none of the interviewees had seen the 14 relevant variables that were distinguished from the literature (chapter 2.10). This was done to identify missing variables and to identify the relevance of these variables. Subsequently, they were asked to rank the 14 different variables from 1 to 14, where 1 was the most important variable and 14 the least important variable. The average of the scores is taken to identify the average relative importance of the 14 different variables by expert opinion. Thereafter, a set of final input variables was determined based on the literature review and interviews. This final set of input variables also consisted of qualitative variables. The ANN model only can handle numerical values, therefore qualitative variables were transformed into quantitative variables. The way in which this is done will be explained in chapter 4.

3.1.2 Collecting and pre-processing data

[redacted]

Lastly, a final database was set up by connecting the database with the output of the survey. After collecting the data, the data was divided into three sets: training, validation, and testing. In this division, the training set is about 70% of the total data set, and the validation and testing set represents 15% of the total dataset each (Hagan et al., 2014). It is important that each set is a good representative of the full data set. The simplest and most common method for data division is to select datasets at random. In addition, it is common to normalize the data before applying them to the network. The purpose of the normalization is to facilitate and enhance network training. In multilayer networks, sigmoid transfer functions are often used in the hidden layers. These functions become saturated when the net input is greater than three and will lead to very small gradients. It is common to normalize the data before applying them to the network. The standard method is to normalize the data so that they fall into a standard range between -1 to 1 (Janssen, 2018). Therefore, this is done for both the input data as output data.

3.2 Training phase

In order to improve the performance of the model, an optimization strategy was developed. This strategy consists of three iterative processes which were carried out sequential. The first iterative process determined the best performing training algorithm and the best model based on the complete dataset. The second iterative process determined the best performing input variables, and therefore the dataset was altered. The last iterative process consisted of finding the range of proposal value wherein the model performed best. In order to develop and train an artificial neural network, a MATLAB script is needed to be established. This is done by using the Neural network Toolbox (Beale, M. H., Hagan, M. T., & Demuth, 2018), this in order to develop the initial script. Subsequently, the script is extended and altered by means of facilitating the optimization strategy. With the use of this script the network is trained, analysed and optimized. First of all the optimization strategy with the coherent three iterative processes are explained.

3.2.1 Optimization strategy

First iterative process

The first iterative process is about determining the best performing training algorithm and the best model based on the complete dataset (see Figure 3-2 below). In this iterative process, three alternative training algorithms are tested. The training algorithms that are described in the literature review in chapter 2.9. In the first iterative process the Levenberg-Marquardt, Bayesian Regularization and Resilient backpropagation training algorithms will be tested. The first iterative process starts with importing the total data set. Subsequently, a training algorithm is selected. Hereafter, the training enters a network architecture optimization module, which is illustrated on the right-hand side of Figure 3-2. Here, the growing method was used. In this technique, the training is started with a single hidden neuron and one neuron is added to the hidden layer every iteration. The training is ceased whenever significant overfitting emerges. For every architecture, the performance of the network is retained. When the network architecture is optimized, the next training algorithm is selected until all the training algorithms are tested. After the first iterative process is finished, the best training algorithm and best network architecture that explains the total dataset is found.

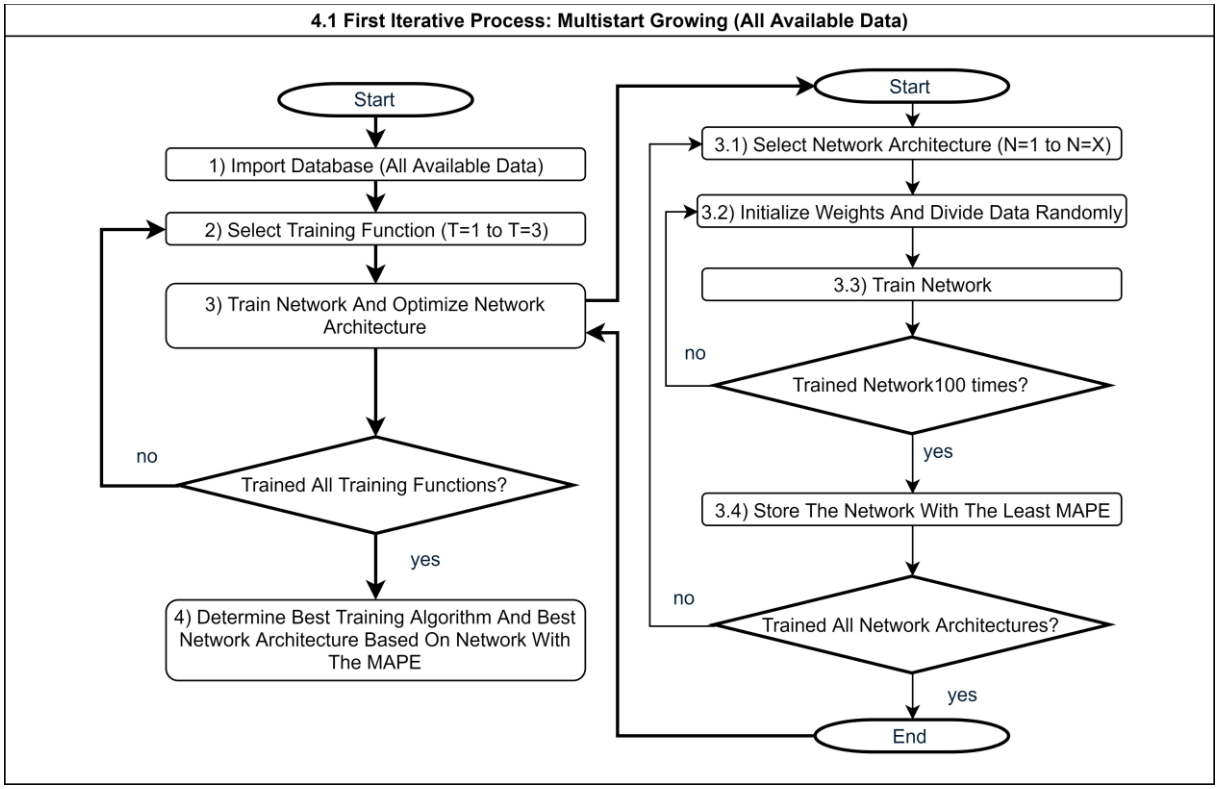


Figure 3-2. Optimization strategy: first iterative process

Second iterative process

After the best training algorithm is found, the network that obtained the highest performance is analysed to determine the relative importance of the input variables. This is the start of the second iterative process (see Figure 3-3 below). In order to find the simplest model that explains the data, it could be helpful to eliminate redundant or irrelevant input variables. By calculating the relative importance of input variables of the network with the highest performance, redundant input variables can be removed and generalization can increase. A method called Connection Weights Algorithm (Olden & Jackson, 2002) can be used to calculate the relative importance of a given input variable of a neural network and can be defined as Equation 12 below. This approach is based on estimates of the network's final weights obtained by training the network (Ibrahim, 2013; Janssen, 2018).

$$RI_x = \sum_{y=1}^{m} W_{xy} W_{yz} \qquad \text{(Equation 12)}$$

The next step is to eliminate the variables that have low impact and retrain the network with the training algorithm that was determined by the highest performance in the first iterative optimization process. The elimination is done by excluding one variable at a time until there is one variable left. Also, the training will be ceased when there is a significant drop in performance when excluding a certain variable. Due to the fact that the number of input neurons decreases, the number of neurons in the hidden layers also potentially need to be changed. Therefore, the strategy of growing is also used again. Eventually, it becomes clear what the simplest model that explains the data is. This model has the best training algorithm, most relevant input variables and the best fitting architecture.

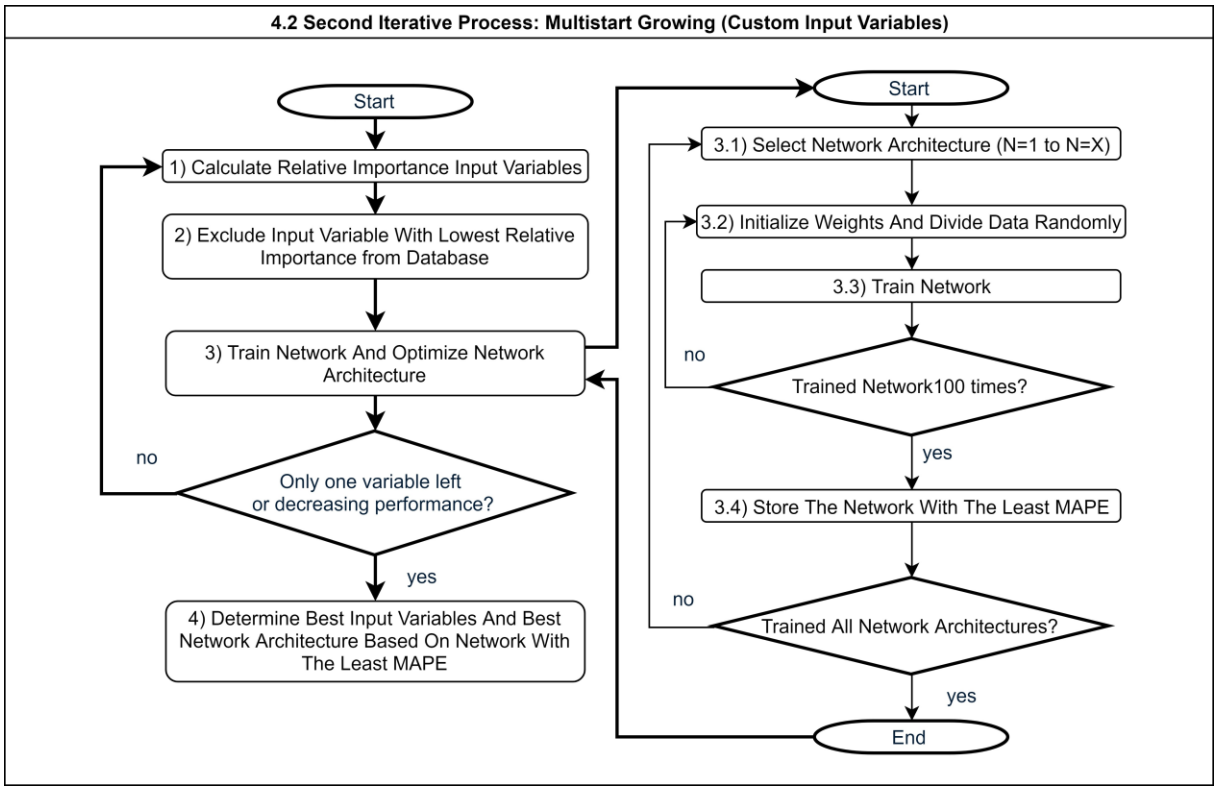


Figure 3-3. Optimization strategy: second iterative process

In addition to the connection weight algorithm for the determination of the relative importance of the input variables, two other methods are used namely; multiple linear regression analysis and expert opinion. MLR analysis is a suitable method to identify which variables have a significant influence on the proposal price. It can help determine whether there is a linear association or causation between the independent variables and proposal price. First, the relative importance of the independent variables is determined by the unit drop in R^2 when a variable is deleted from the sample. R^2 is the coefficient of determination and shows the percentage of variation in a dependent variable which is explained by all the independent variable together. The larger the drop in R^2 when removed from the sample, the more important it is assumed to be. In addition, the data is checked on whether it has multicollinearity. This occurs when two or more independent variables are highly correlated with each other. When collinearity is present, it is hard to find out if one variable causes an effect or the other (van der Steen, 2018). Therefore, when there is multicollinearity in the data, some variables could be redundant and removed. Finally, the last method to

determine the relative importance of the input variables is by expert opinion. As described in the pre-training phase the variables were ranked by experts. This ranking is also used as a way to determine the relative importance of the input variables. Also, neural networks are developed based on the relative importance of the input variables determined by the MLR analysis and expert opinion. The performances of the neural network can be compared with the neural network based on the results from the connection weight algorithm. Based on the comparison it can be known what the best method is to determine the most important variables for a neural network.

Third iterative process
Lastly, neural networks can interpolate accurately throughout the range of the data preceded, however extrapolation outside the range of the training set is of lower quality. There is no way to prevent errors of extrapolation unless the data that is used to train the network covers all regions of the input space where the network is used. In addition, if there is a relatively small number of data points in a specific region, this could also lead to bad interpolation. This is a simple result of not enough data for that specific region, and therefore it cannot be trained properly for that region. In order to ensure preventing bad results from extrapolation, it should be ensured that the network is not used for project values that are outside the dataset on which the network is trained.

In addition, we can exclude certain project value ranges where there are relatively low numbers of examples. This will lead to a smaller range of projects for which the neural network can be used, however, it could lead to a higher performance of the model when interpolating. This is done in the third and final iterative process [redacted] where a selection of project value range is made. In this process, three selections of data regions were made based on the results of the second iterative process. First of all, it was decided to proceed the training with the 5 different network architectures that came out best in the second iterative process. However, when a data selection is made, the complexity of the underlying function of the data could be different compared to the full database. Therefore, the growing technique was used again and the number of hidden neurons was changed for every network in each training set. [redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]

[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

3.3 Post-training phase

Lastly, the post-training phase is about the internal validation of the neural network model and transforming the neural network into a usable application. The internal validation will provide insights into how the model will perform outside the training sample. Therefore, a feeling is acquired for the generalization of the model. In addition, the best performing network based on training algorithm, input variables, architecture and proposal value range will be used to develop an application. After training the neural network with its architecture, weights and biases and transfer functions is saved and acts as a back-end function for the application. This application can then be used in practice for new tenders.

3.3.1 Validation best performing model

When the MATLAB application is developed and installed a validation of the method can be performed. Steyerberg & Harrell (2016) suggest that there are three types of validation techniques namely; apparent, internal and external. Apparent validation concerns the performance of a sample that is used to develop the model. Internal validation concerns the performance of a sample different than the sample on which the model is developed. However, this sample should be similar to the developed sample. External validation is done with the use of new data, that was not available on the time the model was developed. Usually, this data is gathered in a different way by a different researcher or user. In this research, both apparent and internal validation is performed, however, external validation is not carried out.

One of the methods that can be used for internal validation of a predictive model is the split-sample validation technique. The split sample technique is used in this research by dividing the sample into training and test samples. By training the model on the training sample and then test the model using the test sample, the internal validity and performance can be determined. Internal validation of predictive models is important as it provides an honest estimate of the performance that can be obtained for a sample similar to the development sample. Also, it provides an upper limit to what may be expected in other settings (e.g. external validity).

In addition, a common source of variance in a final model is the noise in the training data and the use of randomness in the training phase. The training of a neural network is involved with two stochastic elements, due to which every training run a different performance and different variance will emerge. To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. This was done by training a model several times and evaluate the variance that is introduced by the stochastic elements. This is also called bootstrapping. Bootstrapping is a method for estimating the distribution of an estimator or test statistic by resampling the data or a model estimated from the data (Nanculef & Salas, 2004). Bootstrap plans can be used for estimating the uncertainty associated with a value predicted by a feedforward neural network. By doing so, a more robust estimate of the variance of the model can be acquired.

3.3.2 Develop and deploy MATLAB application

A MATLAB app is a self-contained MATLAB program with a user interface (UI) that automates a task or calculation (Mathworks, 2019). All the operations that are required to complete a certain task are performed within the app. For example, operations are getting data into the app, performing a calculation on data and getting results. By developing an application, the newly developed cost estimation method can also be used as a standalone version. This means MATLAB is not needed to perform cost estimations with the use of the neural network. One of the requirements for developing an application is that the main file must be a function and not a script. Therefore, the script that contains the developed neural network will be transformed into a function. This function is then connected to the developed UI in MATLAB.

4 RESULTS

In this section, the results of the execution of the proposed method are presented. The results are presented in the same order as in the proposed method. First, the results of the pre-training phase are described. In the pre-training phase, the determination of the final input variables is presented. In addition, the results of the collection of data are described. Thereafter, the results of the training phase are described. Here, the results of the development of the neural network model are presented. Lastly, the results of the post-training phase are discussed. In this last part, the internal validation of the model and the practical implementation gets attention.

4.1 Pre-training phase

4.1.1 Input variables

From the literature, a set of 14 factors that influence the actual costs of engineering services was identified (see chapter 2.10). Based on the interviews, these 14 variables were altered and extended to 16 final input variables. In the 6 questions that were asked before the 14 variables from the literature were shown, interviewees usually provide answers that include the top 5-7 variables in Table 4-1 below. Other variables were also mentioned but less frequent. [redacted] The average of the ranking is taken to identify the average relative importance of the 14 different variables by expert opinion (see Table 4-1).

Table 4-1. Results ranking variables by experts

Variables	Average Score
Scale of work	2,8
Project phases	5,0
Project duration	6,0
Scope of work	6,5
Type of work	6,6
Complexity of design	6,8
Quality of information	7,1
Number of project team members	7,7
Collaborating disciplines	8,5
Type of client and requirements	8,9
Market conditions	9,0
Client's attitude towards design changes	9,2
Project manager experience[4]	9,8
Pre-contract design	9,9

Final Input variables

Based on the interviews some changes are made towards the final set of independent input variables. The complexity of the design is disregarded as cost factor as interviewees explained that the complexity of a project usually is determined by all the other factors that were already present. Also, the variable of the project team experienced is changed into project manager experience as this variable is more convenient to measure[4]. Therefore 13 variables from the literature remain and in addition, the interviewees provided insight into three influencing factors. These three additional factors will be explained below. The final input variables with the quantitative scales

[4] Project team experience was changed into project manager experience as a previous study carried out by van der Steen, (2018) shows correlation between costs of services and level of experience of the project manager. Furthermore, project team experience is subjective and harder to measure.

are shown in Appendix B. Finally based on the literature and interviews the final set of 16 independent input variables can be viewed in Table 4-2 below. [redacted]

Table 4-2. Final input variables[5]

No.	Influencing factor	Variable	Available in databases
X_1	Scale of work	Interval	Occasionally
X_2	Project phases	Nominal	Occasionally
X_3	Project duration	Ratio	Not available
X_4	Scope of work	Ordinal	Not available
X_5	Type of work	Nominal	Not available
X_6	Level of experience on clients side	Ordinal	Not available
X_7	Scope definition (quality of information)	Ordinal	Not available
X_8	Number of project team members	Ratio	Not available
X_9	Collaborating Disciplines	Ratio	Available
X_{10}	Type of client and requirements	Ordinal	Not available
X_{11}	Main market type	Nominal	Available
X_{12}	Client's attitude towards design changes	Ordinal	Not available
X_{13}	Project manager experience	Ordinal	Not available
X_{14}	Pre-contract design	Ordinal	Not available
X_{15}	Contract type	Ordinal	Available
X_{16}	Intensity	Ordinal	Available

Level of experience on clients side
This variable concerns the level of experience of the team on the client's side. Several interviewees explained the effect of the level of experience on the client's side. When the project team on the client side is more experienced, the project progresses more fluently. This due to the fact that they usually know what information they need to provide and also what information they want. On the other hand, when the team on the client side is less experienced, this could potentially lead to less fluently project progression. This because, the team could need more time to review documents, ask more questions, provide incorrect information, etc.

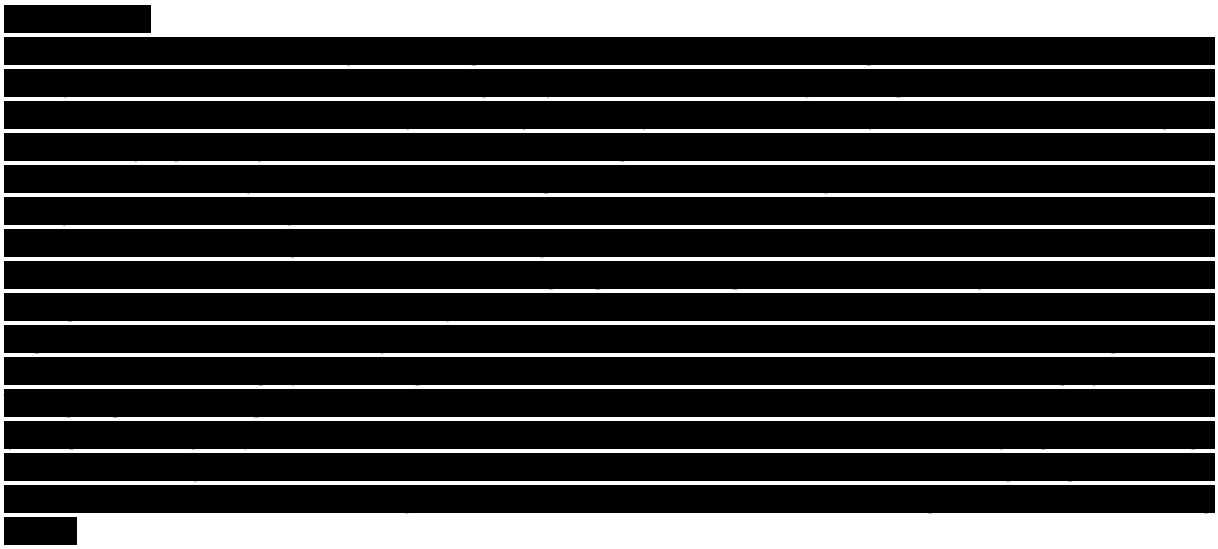
Type of contract
Furthermore, several interviewees described the effect of the contract type on the costs of a project. Generally, there are three types of contract namely; fixed price, reimbursable ceiling and reimbursable no ceiling. A fixed price contract is a type of contract where the payment amount does not depend on resources used or time expended but is lump sum defined. The price that is determined before the project starts will be invoiced to the client. For fixed price projects, the price is usually higher then reimbursable contract types. This due to the increased financial risk that is experienced with a fixed price contract. For reimbursable contracts, the invoiced amount depends on the resources used and time expended. All the costs that are made are invoiced to the client. However, in some situations, a ceiling can be set in order to prevent costs that are too high.

Intensity
Lastly, the intensity of a project is mentioned as an influencing factor. This factor concerns the number of hours/days the team spends on the project on average per week. Some projects are very intense, meaning that a relatively high amount of work is needed to be finished in a relatively low amount of time. This could increase the costs of mistakes and could lower the quality of the work due to increased stress. In contrary when projects are of low intensity, this could mean lower costs of mistakes.

[5] The labeling of the variables can be described as following: not available means that the variables cannot be gathered from the databases, available means that it can be gathered from the databases and, occasionally means that for some projects the data is available. The quantitative scale of the input variables can be viewed in appendix B.

4.1.2 Data collection

The data for projects that were not available in the databases, were collected using an online survey. In this survey, the responsible tender managers or project managers, depending on who made the tender, were asked to provide the missing data of these projects. The survey was set up in the SharePoint environment within the Intranet of Tebodin. The projects that are valid for use in the model are provided with a project ID number. This project ID number could be selected when filling in the survey. This allowed matching the survey results with the already existing database for the projects. The setup of the survey can be found in Appendix A.



[redacted]

4.2 Training phase

Now that the dataset is established the data can be used to train a neural network. As explained in the proposed method, the training phase consists of the initial set up of the model and the execution of the described optimization strategy. First, a selection of training algorithms that are used for training the model was made. Subsequently, a technique for determining the network architecture was chosen. In the first iterative process, the best training algorithm and best network architecture for the total dataset are determined. The second iterative process led to the determination of the most relevant input variables. This was done by calculating the relative importance of the input variables and excluding one variable at a time. Lastly, a selection of project value range was done. By excluding certain project value ranges, interpolation and extrapolation problems were minimized. This was carried out by the third and final iterative process of the optimization strategy. First of all, the results of the first iterative process are described.

4.2.1 Results first iterative process

The first iterative process was about determining the best performing training algorithm and the best model based on the complete dataset. In this process, the best performing training algorithm and architecture is determined. In total three different training algorithms were tested. The training of the model is carried out using a growing technique, which means that the neural network architecture is started with 1 hidden layer and the number of hidden neurons is increased every training run until overfitting occurs. Overfitting occurs when the MSE of the training data becomes very small and the test results are poor.

The best results for the best network architectures for the three different training algorithms are viewed in Table 4-4 below. The results for all the different architectures can be found in Appendix C. In can be concluded that with 16 input variables and the complete dataset, the Bayesian Regularization training algorithm with one hidden layer and 4 hidden neurons performs the best. In this research, the MAPE performance metric is used for selecting the best model when the training of the model is finished. However, the model is only selected if the R values of the training set and test set are relatively similar and close to 1. When this is not the case, the next best MAPE is chosen. Clearly, the correlation coefficient (R) for both the training as testing of the BR-16-4-1 is the closest to 1. Furthermore, the correlation coefficient has the least difference between them. As described in the proposed method, whenever the correlation coefficient is close to one it fits the data well. When the R of the testing and training set are similar this implies a stable model and means that it generalizes well. In addition, Bayesian Regularization has the lowest MAPE score. Which means that it has the lowest mean average percentage error for both the training as the testing set.

Table 4-4. Best results first iterative process[6]

Network Architecture	MSE Train	MSE Test	MSE All	R Train	R Test	R All	MAPE Train	MAPE Test	MAPE All
LM-16-6-1	3.84+08	1.34+12	3.68+11	0.9997	0.9168	0.9684	57.05%	100.18%	77.98%
BR-16-4-1	9.96+07	1.58+11	2.41+10	0.9998	0.9645	0.9796	37.25%	50.36%	39.24%
RP-16-6-1	2.00+9	1.39+11	1.39+11	0.9966	0.7509	0.8666	59.48%	88.68%	89.51%

In order to substantiate the results of the three different training algorithms, the regression plot of all the three best performing networks is shown below. In Figure 4-1 below the regression plot for a neural network trained with the Levenberg-Marquardt training algorithm and 6 hidden neurons is shown. In Figure 4-2, the regression plot for the neural network that was trained with the Bayesian Regularization training algorithm is shown. Lastly, the regression plot for the neural network that was trained with the Resilient Backpropagation algorithm is shown in Figure 4-3 below. In these regression plots, on the X-axis the targets of the dataset (real proposal value) are shown, and on the Y-axis the model outputs (estimated value) are shown. For every training algorithm, four different plots are shown, these are associated with the deviation of the dataset in train-, test- and validation sets. The blue line represents the performance on the training set, the green line represents the performance on the validation set, the red line represents the performance on the test set and the grey line represent the performance in the total dataset.

It is clear that for all the training algorithms the neural network adopted the training data very well, however, results on the test set are not very promising. However, the test results that are most promising are for the Bayesian Regularization training algorithm. The regression line for the test set is closest to the 45-degree line and the individual test results are closest to the 45-degree line compared to the other training algorithms. To conclude, the Bayesian Regularization training algorithm provides the most promising results and is used in the continuation of the optimization strategy.

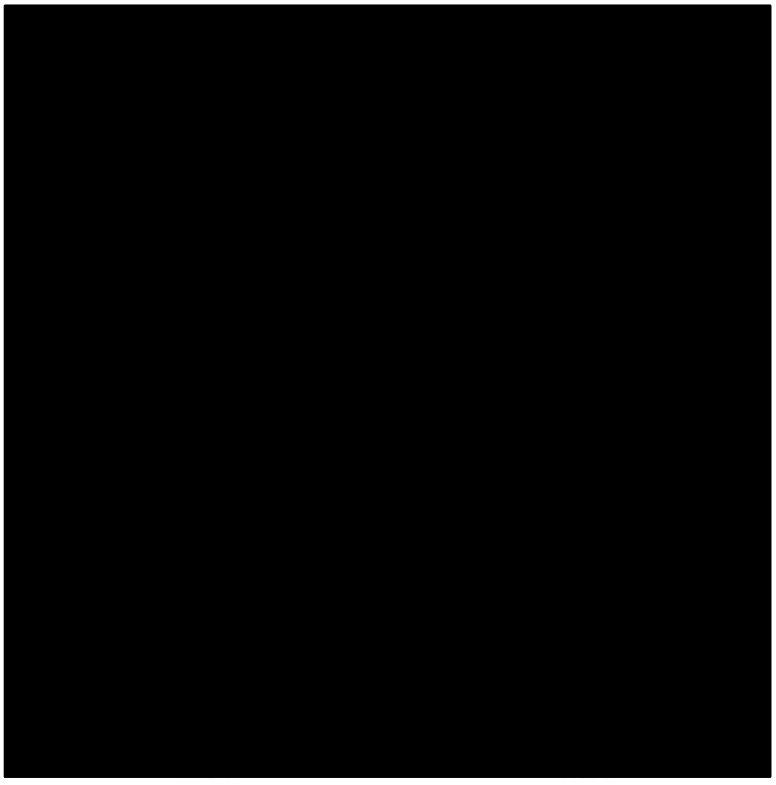
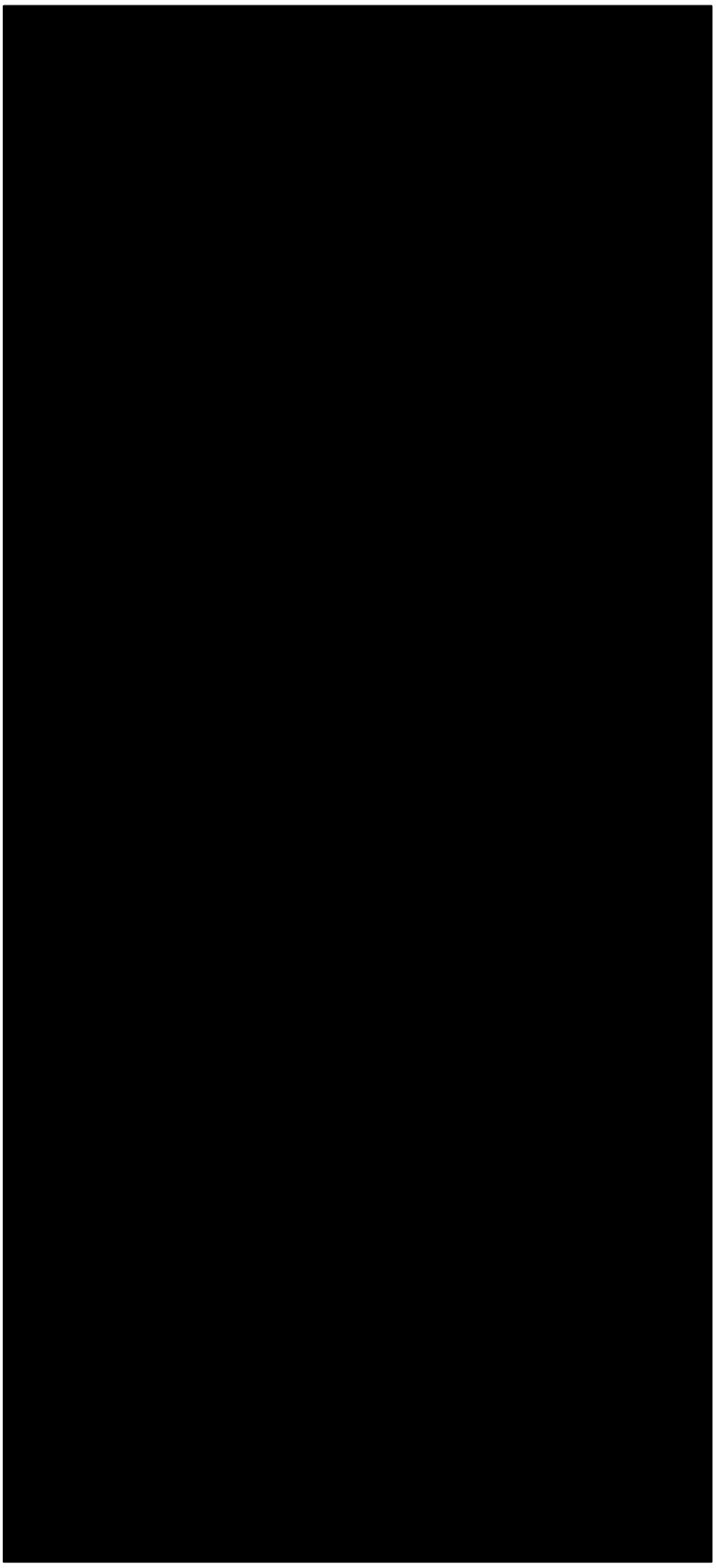


Figure 4-1. Regression plot LM-16-6-1

[6] In the table the network architecture is described as “Name of Training algorithm”-“Number of input Variables”-“Number of hidden neurons”-“Number of output Variables”. Wherein LM stands for Levenberg-Marquardt, BR stand for Bayesian Regularization, and RP stands for Resilient Backpropagation.



4.2.2 Results second iterative process

The second iterative process improved performance by determining the most relevant and most influencing input variables. This was done by calculating the relative importance of the input variables for the neural network that was determined in the first iterative process. It could be helpful to eliminate redundant or irrelevant input variables, to find the simplest model that explains the available data. When eliminating redundant input variables the neural network becomes less complex and therefore, generalization could improve. The elimination is done by excluding one variable at a time until there is one variable left. Due to the fact that the number of input neurons decreases, the neurons in the hidden layers also potentially need to be changed. Therefore, the strategy of growing (see chapter 2.9.2) is also used again. For the determination of the relative importance of the input variables, three different methods are used namely; connection weight algorithm, multiple linear regression analysis and expert opinion. First, the results for the connection weight algorithm method are given.

Connection Weight Algorithm
The neural network that performed the best in the first iterative process is used to calculate the relative importance of the independent input variables. In Figure 4-4 below the relative importance of the input variables for the best performing network are shown. In this figure, it is shown that the scope definition has the lowest relative importance compared to other variables. Therefore, the scope definition was excluded from the sample and training was continued.

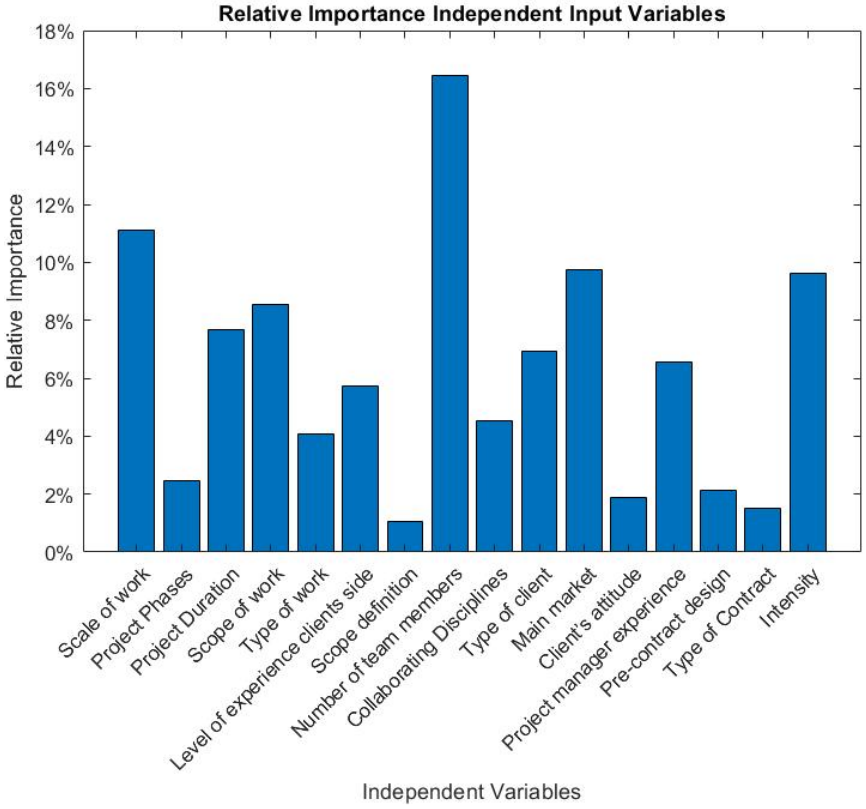


Figure 4-4. Relative importance independent input variables (BR-16-4-1)

When the model was trained with only four variables, the performance significantly decreased. Therefore, the training was ceased after four variables were left. The best results for the best performing input variables and the coherent network architectures can be viewed below in Table 4-5. The results for all the different architectures can be found in Appendix C. The best performance occurred when the ANN was trained using the set of variables that range between the most important 9 variables and 5 variables. The best test performance of the network (MAPE) with all 16 variables was around 50%. While using, 5 variables the test performance has reached 27,41%. Therefore, the performance of the model has significantly increased as variables are selected by calculating the relative importance.

Table 4-5. Best results second iterative process

Network Architecture	MSE Train	MSE Test	MSE All	R Train	R Test	R All	MAPE Train	MAPE Test	MAPE All
BR-9-5-1	5.21+08	1.36+11	2.11+10	0.9992	0.9640	0.9813	48.27%	51.32%	48.73%
BR-8-6-1	6.36+08	1.16+10	2.30+09	0.9995	0.9556	0.9979	55.07%	42.26%	53.13%
BR-7-6-1	7.08+08	3.06+10	5.24+09	0.9994	0.9648	0.9952	46.73%	37.41%	45.32%
BR-6-8-1	3.24+08	9.01+09	1.64+09	0.9997	0.9460	0.9985	35.19%	32.83%	34.83%
BR-5-7-1	4.79+08	3.86+09	9.91+08	0.9996	0.9952	0.9991	33.15%	27.41%	32.28%

The best performing network on all the data was a network with 5 input variables and 7 hidden neurons. To substantiate the performance of this model, the regression plot is shown in Figure 4-5 below. In addition, the relative importance of the input variables is shown in Figure 4-6 below. When the regression plot is analysed, it can be seen that the testing results are very promising. The predictions are concentrating around the optimum 45-degree regression line, however, not perfectly. Furthermore, the testing R and the training R are very close to each other which implies a model that is capable of good generalization. In the relative importance bar chart, it is shown that the project duration is the most important input variable, followed by respectively the number of team members, collaborating disciplines, intensity and scale of work.

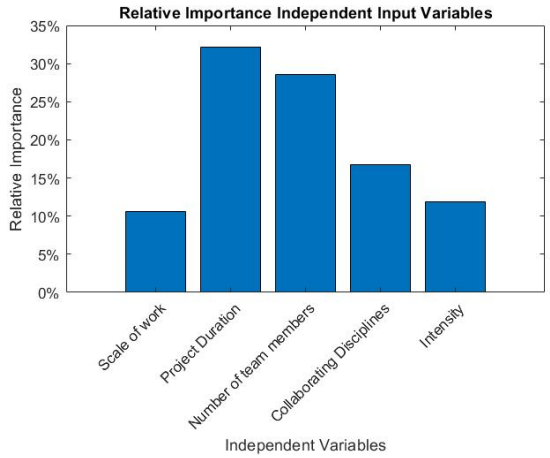


Figure 4-6. Relative importance bar chart BR-5-7-1

However, when the relative error of the estimates are analysed (Figure 4-7) the performance of the model is not optimal. In the figure below, the distribution of the relative error for the entire dataset and the test set is represented. Although the mean absolute percentage error of the total set is 32.28%, a widespread in the relative error for the data points occurred. Around 57% of the overall predictions have a relative error of more than 10%. In addition, 24% of the overall predictions have a relative error larger than 35%. For the test set, 65% of the predictions have a relative error of more than 10%. Furthermore, 35% of the test predictions have a relative error larger than 35%. Therefore, the results of the model are inherent with large variances in the relative error of the predictions.

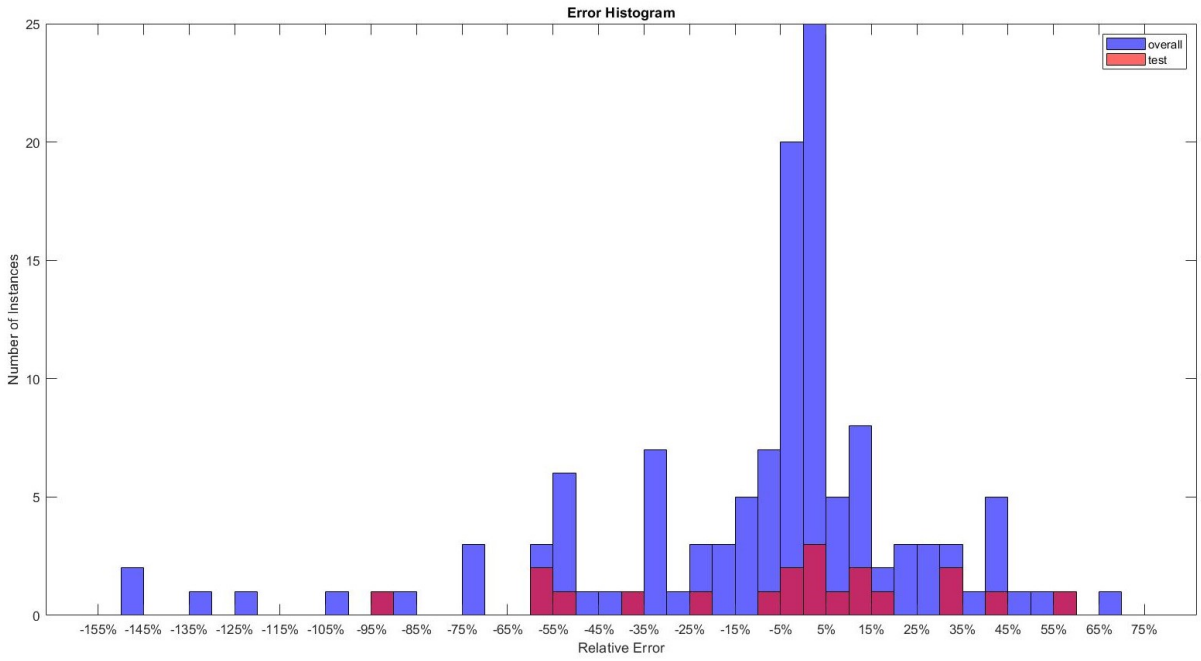


Figure 4-7. Error histogram, with bin sizes of 5%, for BR-5-7-1 with 132 data points

Multiple Linear Regression Analysis

Subsequently, the neural network is trained using the most important input variables that are determined by a MLR analysis. This will allow showing what method for determination of the most important variables gives the best results. First of all, the model fit was examined with the help of Table 4-6 below. The coefficient of determination (R^2) tells how much of the variance in proposal value can be explained by the independent variables in the model. In this case, 84,3% of the variance can be explained by the model. However, this value only applies to the sample. To know what it would be for the population the adjusted R^2 is used. In the population, 78,5% of the variances can be explained. This means that there are other variables that were not included in the model that explains the variance. In other words, the model is not fully complete.

Table 4-6. Model summary of the multiple regression model

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,918a	,843	,785	120497,6194953

a. Predictors: (Constant), Intensity, ClientsAttitude, TypeOfContract, PreContractDesign, ProjectManagerExperience, ScopeOfWork, TypeOfWork, ProjectDuration, Disciplines, MainMarket, ProjectTeamMembers, TypeOfClient, ProjectPhases, LevelOfExperienceClient, ScaleOfWork, ScopeDefinition

Subsequently, to find out which independent variables add significantly to the model the p-value is evaluated. Whenever the p-value is lower than the significance level of 5% it can be stated that the concerning variable significantly predicts the proposal value. In this case, the multiple regression analysis showed that there is statistical evidence that project duration, project team members, disciplines, type of contract and intensity have a significant influence on the proposal value. The significance factor should be lower than 0,05 for it to be significant, the variables that match this criterion are marked grey in Table 4-7 below.

Table 4-7. Coefficients and significance of the independent variables.

	Unstandardized Coefficients		Standardized Coefficients			Correlations		
Variables	**B**	**Std. Error**	**Beta**	**t**	**Sig.**	**Zero-order**	**Partial**	**Part**
(Constant)	-460357.039	170149.968		-2.706	0.010			
ScaleOfWork	21.377	74.064	0.025	0.289	0.774	0.411	0.044	0.017
ProjectPhases	16097.335	16247.190	0.082	0.991	0.327	-0.048	0.149	0.060
ProjectDuration	6544.496	969.518	0.487	6.750	**0.000**	0.449	0.717	0.407
ScopeOfWork	-29467.419	23832.871	-0.093	-1.236	0.223	0.122	-0.185	-0.075
TypeOfWork	-143.464	14450.257	-0.001	-0.010	0.992	-0.087	-0.002	-0.001
LevelOfExperience Client	29290.078	20210.121	0.123	1.449	0.155	0.005	0.216	0.087
ScopeDefinition	4679.353	26325.018	0.022	0.178	0.860	-0.123	0.027	0.011
ProjectTeamMembers	3344.488	1577.644	0.166	2.120	**0.040**	0.120	0.308	0.128
Disciplines	30713.465	6156.744	0.386	4.989	**0.000**	0.583	0.605	0.301
TypeOfClient	5646.161	23142.959	0.021	0.244	0.808	-0.013	0.037	0.015
MainMarket	24143.478	13144.188	0.147	1.837	0.073	0.223	0.270	0.111
ClientsAttitude	-3992.946	14615.954	-0.019	-0.273	0.786	-0.146	-0.042	-0.016
ProjectManagerExperience	35152.389	19556.779	0.140	1.797	0.079	0.469	0.264	0.108
PreContractDesign	-54862.017	28506.616	-0.250	-1.925	0.061	-0.159	-0.282	-0.116
TypeOfContract	-49250.798	24246.116	-0.165	-2.031	**0.048**	-0.226	-0.296	-0.123
Intensity	115153.627	18768.581	0.456	6.135	**0.000**	0.457	0.683	0.370

a. Dependent Variable: ProposalValue

In addition, the relative importance of the independent variables is determined by the drop in R^2 when the variable is removed from the sample. The R^2 value determines how much of the variance is explained by the model. Therefore, when an independent variable is removed from the sample, the drop in R^2 determines how much of the variance is explained by the variable. The assumption here is, the more the drop in R^2, the more important the independent variable is. The drop in R^2 when removed from the sample can be calculated by squaring the part correlation of an independent variable. In Table 4-8 below the relative importance according to the drop in R^2 calculated by the MLR model is shown. It becomes clear that the 5 variables that were statistically significant also occupy the top 5 in the relative importance by a drop in R^2.

In addition, there was one case of possible multicollinearity in the data. Hair, Black, Babin, & Anderson, (2014) suggest that researcher always should asses the degree and impact off multicollinearity when Variance Inflation Factor (VIF) values of 3 to 5 are present. Two variables match this criterion namely ScopeDefinition with a VIF of 4,149 and PreContractDesign with a VIF score of 4,630. The Pearson correlation of these two independent variables is 0.807, and therefore suggest a high amount of "shared" variance. Therefore these are not used simultaneously in the ANN model.

Table 4-8. Relative importance independent variables MLR

Variables	Part Correlation	Squared Part Correlation	Relative Importance
ProjectDuration	0.407	0.1656	1
Intensity	0.37	0.1369	2
Disciplines	0.301	0.0906	3
ProjectTeamMembers	0.128	0.0164	4
TypeOfContract	-0.123	0.0151	5
PreContractDesign	-0.116	0.0135	6
MainMarket	0.111	0.0123	7
ProjectManagerExperience	0.108	0.0117	8
LevelOfExperienceClient	0.087	0.0076	9
ScopeOfWork	-0.075	0.0056	10
ProjectPhases	0.06	0.0036	11
ScaleOfWork	0.017	0.0003	12
ClientsAttitude	-0.016	0.0003	13
TypeOfClient	0.015	0.0002	14
ScopeDefinition	0.011	0.0001	15
TypeOfWork	-0.001	0.0000	16

In Table 4-9 below the best results of the training based on the relative importance of the MLR are viewed. All the results for the training process based on the MLR can be viewed in Appendix C. In this training process, the most important independent input variables that are shown in Table 4-8 are used. The best network test performance of 42,47% MAPE was achieved with 7 hidden neurons. This was the case when the 5 most important independent variables distinguished by the MLR analyses were used. Therefore, the most important variables that are determined by the MLR analysis do not perform better while used in training a neural network, compared to the connection weight algorithm method. Multiple linear regression determines linear relationships between independent variables and a dependent variable. Some of the variables could have a non-linear relationship with the dependent variable. Non-linear relationships can be determined by ANNs, however, not by means of a MLR analysis. Therefore, this could mean that the most important variables that were determined by the connection weight algorithm perform better due to the identification of non-linear relationships.

Table 4-9. Best results ANN based on MLR

Network Architecture	MSE Train	MSE Test	MSE All	R Train	R Test	R All	MAPE Train	MAPE Test	MAPE All
BR-7-6-1	3.16+08	4.20+10	8.53+09	0.9998	0.8419	0.9921	46.68	52.70	47.87
BR-6-6-1	4.07+08	3.14+10	6.51+09	0.9996	0.9784	0.9939	38.83	42.56	39.56
BR-5-7-1	1.27+08	1.06+11	2.09+10	0.9999	0.9065	0.9806	23.56	42.47	27.28

Expert opinion

Subsequently, the neural network is trained using the most important input variables that are determined by expert opinion. In the results of the interviews, it became clear what the most important input variables were. In Table 4-10 below the 7 most important variables determined by the experts are shown. These variables were used in the neural network.

Table 4-10. Top 7 ranking variables by experts

Variables	Ranking
Scale of work	1
Project phases	2
Project duration	3
Scope of work	4
Type of work	5
Quality of information	6
Number of project team members	7

In Table 4-11 below the best results of the training based on the relative importance by expert opinion are viewed. All the results for the training process based on the expert opinion can be viewed in Appendix C. The best network test performance of 93,25% MAPE was achieved with 5 hidden neurons. This was the case when the 5 most important independent variables distinguished by the MLR analyses were used. Therefore, the most important variables that are determined by the expert opinion do not perform better while used in training a neural network, compared to the connection weight algorithm method and MLR analysis

Table 4-11. Best results ANN based on Expert Opinion

Network Architecture	MSE Train	MSE Test	MSE All	R Train	R Test	R All	MAPE Train	MAPE Test	MAPE All
BR-7-2-1	5.05+10	5.39+10	5.12+10	0.9583	0.9449	0.9519	167.69	121.04	158.50
BR-6-7-1	2.06+10	9.90+10	3.60+10	0.9824	0.9405	0.9668	210.61	105.18	189.84
BR-5-6-1	2.17+11	2.65+11	2.27+11	0.7530	0.8609	0.7664	274.98	93.25	239.19

4.2.3 Results third iterative process

The third iterative process is established and performed to minimize the error resulting from interpolation. Neural networks generally can interpolate accurately through the range of the data preceded. However, this is only the case if enough data is present for certain data ranges. Whenever there is a relatively small number of data points in a specific project value range, interpolation could lead to bad results. In addition, when the neural network is used for project values that are outside the dataset, there are high chances of bad results. Neural networks are generally bad in extrapolation. Therefore, it is advised that the neural network is not used for data ranges outside the training set. To minimize the effect of bad interpolation, certain project value ranges can be excluded. For example, when there are relatively low numbers of data points is a specific area it could be beneficial to exclude these data points, to improve the performance of the model.

In Table 4-12 below the best results of the third iterative process are viewed. All the results for the third iterative process can be viewed in Appendix C. The best network architecture per data selection is shown. For all the 3 data selection sets the performance improved with regard to the second iterative process. However, the best results occurred with the 2nd data selection dataset. [redacted] With these settings, a test performance of 13,65 % MAPE was achieved. Very important is that the R-value for both the training and test set are very similar and only differ by 0.0013. Furthermore, the R values for both sets are very close to 1 and therefore indicate a perfect fit. In addition, the performance for the training MAPE, test MAPE and overall MAPE are very similar.

Table 4-12. Best results third iterative process

Network Architecture	MSE Train	MSE Test	MSE All	R Train	R Test	R All	MAPE Train	MAPE Test	MAPE All
BR-5-7-1 (1th selection)	1.11+08	1.44+10	2.29+09	0.9989	0.9705	0.9819	24.0358%	23.6123%	23.9712%
BR-7-4-1 (2nd selection)	5.49+08	8.77+08	6.15+08	0.9957	0.9944	0.9954	13.6477%	13.6481%	13.6478%
BR-5-6-1 (3rd selection)	2.12+08	1.81+09	4.63+08	0.9915	0.9581	0.9832	16.2460%	21.8696%	17.1297%

4.3 Post-training phase

In the post-training phase, the internal validation of the best performing neural network from the training phase is carried out. The internal validation will provide insights into how the model will perform outside the training sample. Therefore, a feeling is acquired for the generalization of the model. This will lead to the answer to the 5th research question. In addition, a MATLAB function is generated from the best performing neural network. This function incorporates the neural network and allows to use the neural network for other proposal estimates. The function is linked to a UI and subsequently exported as an application. With the use of this application, external model validation can be carried out based on datasets that are outside the development datasets. In addition, the neural network can be used for application on real-world projects.

4.3.1 Best performing neural network

The proposed optimization strategy lead to the determination of the best neural network based on best training algorithm, most important input variables, best-performing architecture and a selection of proposal value range. The best performing network based on these criteria is a network with 7 input variables and 4 hidden neurons in one hidden layer. [redacted] To substantiate the performance of this model, the regression plot is shown in Figure 4-8 below. In addition, the relative importance of the input variables is shown in Figure 4-9. The regression plot shows that both the training as testing results are very promising. For the test results, the regression line is perfectly in line with the optimum 45-degree regression line. In the relative importance bar chart, it is shown that the intensity is the most important input variable, followed by respectively the number of team members, project duration, collaborating disciplines, type of contract, project phases and scale of work.

[redacted]

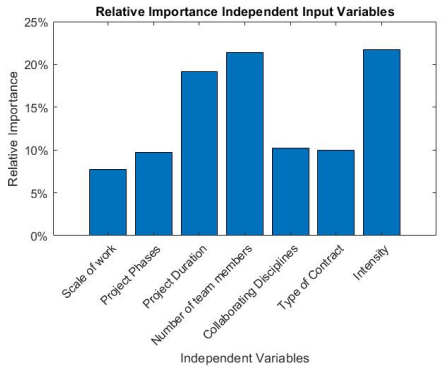


Figure 4-9 Relative importance bar chart BR-7-4-1

[redacted]

The relative error of the individual estimates are shown in Figure 4-11 below. In the histogram, it can be seen that more of the predictions concentrated around the 0% error baseline. For this model, the mean absolute percentage error of the total set is 13,65%. In this case, 38% of the overall predictions have a relative error of more than 10%. For the test set, about 33% of the predictions have a relative error of more than 10%. Furthermore, 12% of the overall predictions have a relative error of more than 35%. For the test set, about 8% of the predictions have an error larger than 35%. This means that compared to the best model from the second iterative process, the variance in the relative error is lower. 37% of all the predictions, both on training as on test data, have a relative error of less than 5%. Eventually, the mean absolute percentage error (MAPE) for the test predictions is 13,65% with a maximum error of 62% and a minimum error of 0,32%.

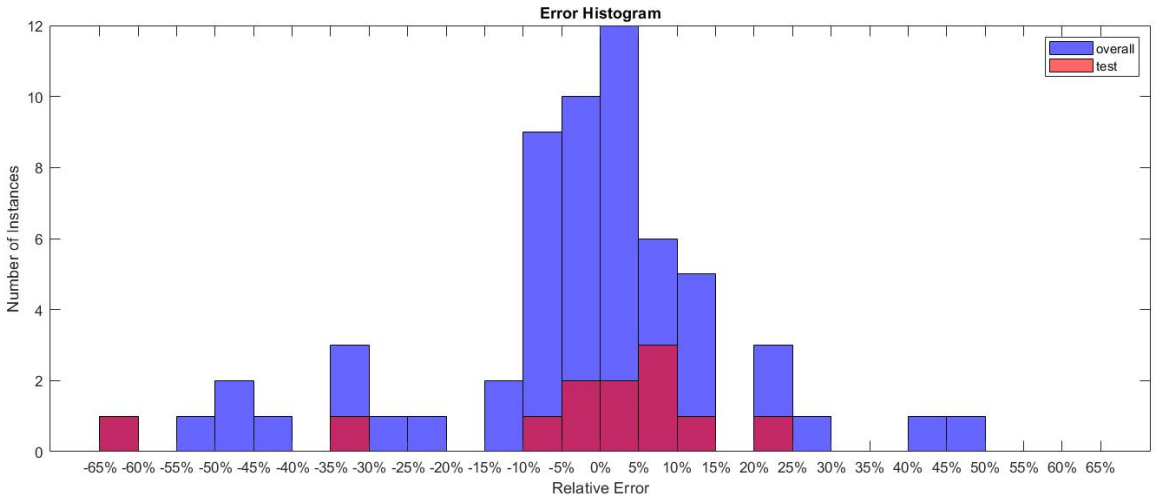


Figure 4-11. Error histogram, with bin sizes of 5%, for BR-7-4-1 with 60 data points

[redacted]

Bootstrapping

Although the mean absolute percentage error of the test prediction is relatively low, the problem with the final model, as with most final models, is that they suffer variance and uncertainty in their predictions. It is this final model, that will be used to make predictions on new data where the outcome is not known. A common source of variance in a final model is the noise in the training data and the use of randomness in the training phase. As explained in chapter 2.9, the training of a neural network is involved with two stochastic elements, due to which every training run a

different performance will emerge. The first stochastic element regards the initialization of weights and biases, which is done randomly every training run. The second stochastic element is the random division of training, testing and validation set. Since there is no one best split of the data or obvious choice for the initial weights etc., many realizations are drawn in order to understand the impact of these choices (Lebaron & Weigend, 1995). To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. A more robust approach is to repeat the experiment of evaluating a stochastic model multiple times. Bootstrapping is a method for estimating the distribution of an estimator or test statistic by resampling the data or a model estimated from the data (Nanculef & Salas, 2004). Bootstrap plans can be used for estimating the uncertainty associated with a value predicted by a feedforward neural network.

The two types of variances can be measured in the final model. The variance introduced by the stochastic nature of the algorithm (random weight initialization) can be measured by repeating the evaluation of the algorithm on the same training dataset and calculating the standard deviation of the estimates. The variance introduced by the randomly selected training data can be measured by repeating the evaluation of the algorithm on different samples of training data and then calculate the mean and standard deviation of the estimates. Often, the combined variance is estimated by evaluating several models that are developed both with random initialization of the weight and random division of the datasets. In this research, the model with 7 input variables 4 hidden neurons was trained 100 times with different initialized weights and random division of data. For all the 100 models, the performance was analysed. For every model, the MAPE test performance was calculated (see Appendix D). Subsequently, the mean MAPE and the standard deviation were calculated for all the models combined. The average MAPE is 61,73% with a standard deviation of 31,27%. Therefore, the more robust estimate of the MAPE of the model is larger compared to the final optimal model.

For every network 12 test prediction are provided, so in total 1200 test prediction are provided. The percentage error of these test predictions are shown in Figure 4-13 below. When all the testing results of the 100 training runs are combined in one grand mean, a value of -23.25% average error can be identified. This means that the model on average tends to predict a smaller value than the actual target. In this case, the standard deviation is 111,43%. Therefore, while the final model has reasonable accuracy, the model is perceived as very unstable when the additional source of variance due to the stochastic nature of the model is taken into account.

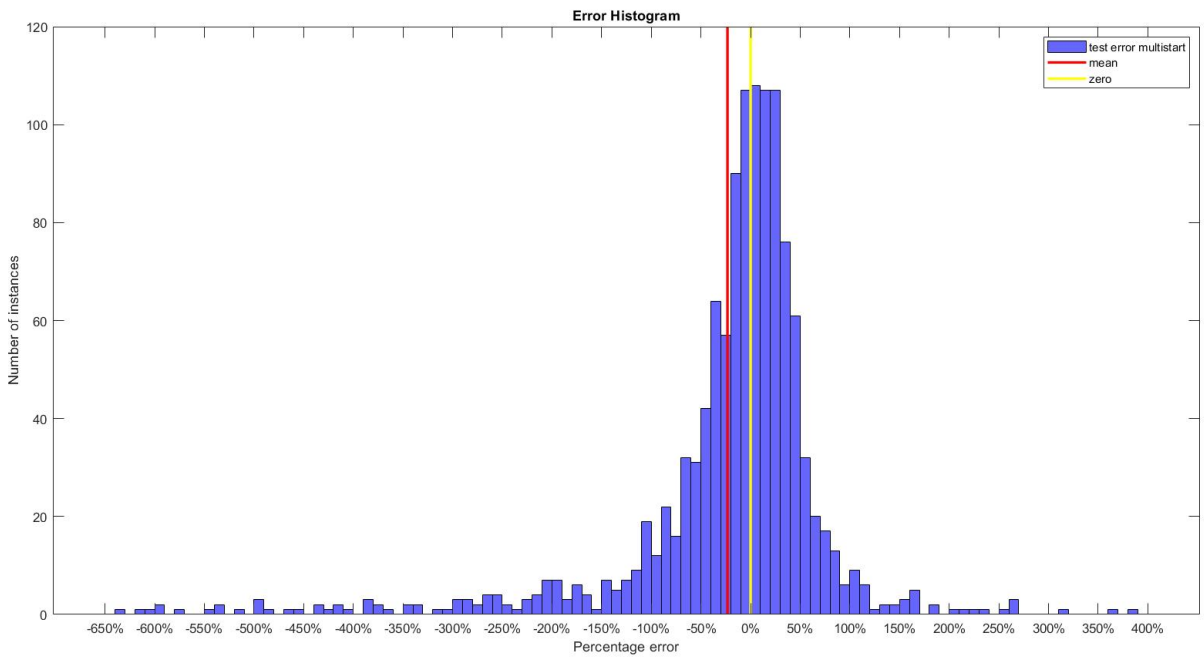


Figure 4-13. Error histogram, with bin sizes of 10%, for 100 x multistart with 60 data points

Lastly, the results provide the relative importance of the independent input variables that are distinguished by the optimization strategy. Here, the top seven independent variables are determined by the relative importance calculation based on the best performing neural network. The sequence of the other variables is determined by the order in which they were excluded in the second iterative process of the optimization strategy.

Table 4-14. Relative importance independent variables optimization strategy

Variables	Relative Importance
Intensity	1
ProjectTeamMembers	2
ProjectDuration	3
Disciplines	4
TypeOfContract	5
ProjectPhases	6
ScaleOfWork	7
MainMarket	8
PreContractDesign	9
LevelOfExperienceClient	10
ClientsAttitude	11
TypeOfWork	12
TypeOfClient	13
ScopeOfWork	14
ProjectManagerExperience	15
ScopeDefinition	16

[redacted]

[redacted]

[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]
[redacted]	[redacted]

5 DISCUSSION

5.1 Discussing results

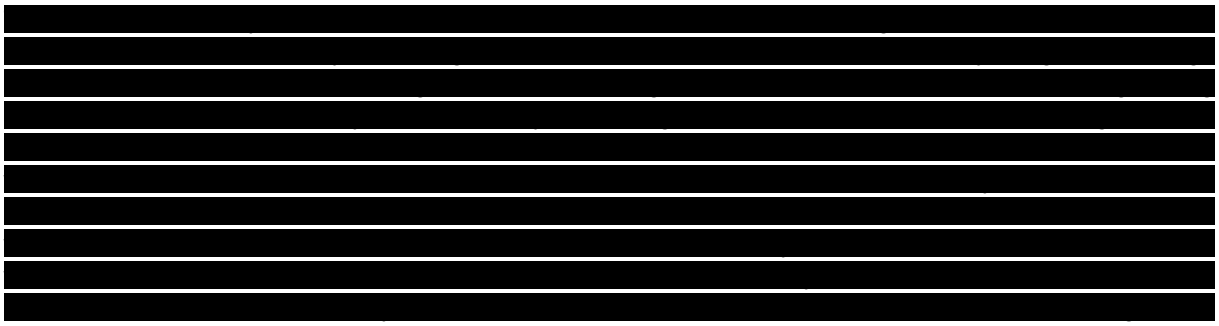
Due to the fact that the ANN model is based on the proposal value or the predictions that were made using the detailed estimation method, the accuracy is limited to the accuracy of the proposal values. This means that no statements can be made on how well the new method performed with respect to the actual costs of a project. However, the performance of the ANN model can be compared with the currently used cost estimation method. The performance of the model was determined by using the internal validation method split sample. The ANN model has an average relative error of 13,65% with respect to the currently used estimation method. With an average accuracy of 86,4% based on 12 individual test cases, the model is fairly accurate with respect to the accuracy that is obtained with the currently used estimation method. In two test cases, the model predicted the costs of engineering services with an accuracy of 99,7%. In fact, 50% of the testing cases were predicted with an accuracy of 94% or higher. However, in one case the model predicted the costs with an accuracy of only 38%. Furthermore, in two other cases, the model obtained a low accuracy of 68,8% and 76% respectively. Eventually, the mean error for the test predictions is 13,65% with a maximum error of 62% and a minimum error of 0,32%. Therefore, while the average accuracy of the testing results is relatively high, the deviation of the individual predictions is still high.

In addition, the training of a neural network is involved with stochastic elements, due to which every training run a different performance and different variance will emerge. To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. Based on the prediction of 100 different networks, The average MAPE is 61,73% with a standard deviation of 31,27%. Therefore, the more robust estimate of the MAPE of the model is larger compared to the final optimal model. Therefore, while the final model has reasonable accuracy, the model is perceived as very unstable. This is identified by taking the additional source of variance due to the stochastic nature of the model into account. Therefore, implementing this method in practice should be considered carefully and is not advised at this moment.

The accuracy that is obtained with the ANN model is considered as average when compared to other ANN cost estimation methods within the construction industry (see Table 5-1 below). In other ANN cost estimation methods, errors of 10,4%, 7%, 16,6%, 4%, 17%, and 6,2% were achieved. However, these results were achieved for estimated costs for contractors and not for engineering services. The study performed by Hyari et al. (2016) is the only study that could be identified that also estimated the cost of engineering services within the construction industry using ANNs. In their study, they revealed an error of 28,2%. Therefore, the results that are achieved in this study are showing a significantly better performing model. In the study performed by Hyari et al. (2016) the cost influencing factors were determined by interviewing experts and showing them the available data. Therefore, in their research, they use only 5 variables that were available and could be quantified.

Table 5-1. Comparison with earlier work

No. of data points	Performance	Sources:
28	10,4% (MAPE)	(Cheng et al., 2010)
30	7% (MAPE)	(Günaydin & Doğan, 2004)
224	28,2% (MAPE)	(Hyari et al., 2016)
288	16,6% (MAPE)	(Emsley et al., 2002)
71	4% (MAPE)	(Mohammed Arafa and Mamoun Alqedra, 2011)
52	17% (MAPE)	(Mahamid, 2013)
813	6,2% (MAPE)	(Arage & Dharwadkar, 2017)



[redacted]

5.2 Limitations research

[redacted]

[redacted]

In addition, in this research, no effort was made towards the external validation of the model. External validation concerns the performance of the predictive model on a new sample, different from the development sample. This set is a fully independent external data set that is not available at the time of development of the prediction model (Steyerberg & Harrell, 2016). External validation can be a very good test of generalizability and applicability in practice. Reasons for assessing performance in other datasets include quantifying optimism from model overfitting or deficiencies in statistical modelling during model development (e.g. small sample size, inappropriate handling of missing data) (Collins et al., 2014). Also, there could be different interpretations of the definitions of the variables that are used in the model. This could mean that the testing results from the internal validation are optimistic.

Furthermore, the model developed in this research is considered improvement with regard to the increase in the pace of preliminary cost estimation in engineering consultancy firms. However, this could not be proven within the research timeframe. In the interviews, some participant explained that they could provide the information for the variables that were determined for new projects within an hour after reading a RFQ. This would be a major improvement as the current estimation method sometimes takes days or even weeks. However, this could not be proven as no test or external validation was performed.

6 CONCLUSIONS

6.1 Conclusion

The goal of this research was to use the existing tacit knowledge in data about past projects to perform cost estimation on new projects by developing an accurate AI-based cost estimation method. This was done by providing an answer to the following research question: How can an accurate AI cost estimation method be developed, to help engineering consultancy firms utilize the existing tacit knowledge that is captured in data to improve speed when estimating costs of engineering services in the tender phase? In this research, a neural network was developed that can estimate the preliminary costs of engineering services based on 7 independent variables. The results showed that artificial neural networks (ANNs) can obtain a fairly accurate cost estimate, even with small datasets. The method led to a neural network consisting of a seven-neuron input layer, a four-neuron hidden layer that used sigmoid transfer functions and a linear single-neuron output layer.

This study investigated the problems regarding currently used cost estimation methods. Found was that these methods do not have the capacity to fully utilize the existing tacit knowledge about past projects and their estimated costs. Therefore, estimation methods tend to be slow [redacted]

Subsequently, investigated was how the problems regarding the currently used estimation problems could be solved using modern AI-based cost estimation methods. Findings in the literature review revealed that artificial neural networks (ANNs) have the potential to overcome the problem.

Hereafter, the cost components that affect the costs of engineering services were identified by a literature review and interviews with experts. This lead to the findings of 16 different variables that could potentially influence the proposal price within a tender. Not all the data that was needed for the model development was available in the databases and a survey was carried out to gather the missing data. Eventually, the data of 132 projects were gathered and was valid for the use in the ANN.

Subsequently, a method was established to develop an ANN and to improve its performance. By developing this method, assumptions were made based on the literature review. The results build on existing evidence of the principles behind creating a neural network that can generalize well proposed by Hagan et al. (2014). The first assumption was that different training algorithms resulted in different performances of the model. The method affirmed this assumption and this study showed that a network trained with the Bayesian Regularization provided the best performance. The second assumption was that excluding potential redundant input variables would increase the performance of the neural network. The proposed method in this study confirmed this assumption and provided evidence that supports this hypothesis. In fact, using 7 of the 16 input variables led to the highest prediction performance of the model. The determination of the most important and relevant input variables was done most successful with the use of the connection weight algorithm. The most relevant input variables that influence the proposal price that were discovered were: project duration, project team members, number of disciplines, intensity, project phase, type of contract and scale of work. [redacted]

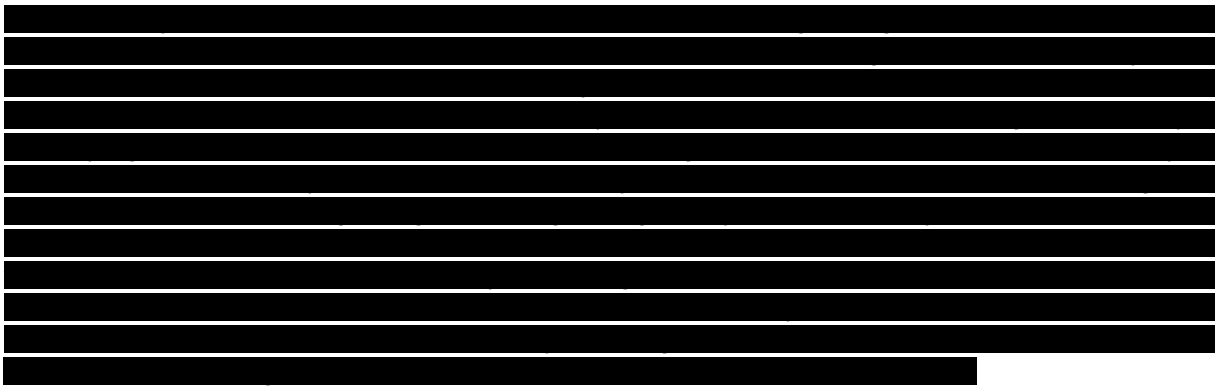
Eventually, the results showed that artificial neural networks (ANNs) can obtain a fairly accurate cost estimate quickly, even with small datasets (60 data points). The average accuracy that is obtained is 86,35% or an average relative error of 13,65% with respect to the results obtained from the current estimation method. Due to the fact that the ANN model is based on the proposal value or the predictions that were made using the detailed estimation method, the accuracy is limited to the accuracy of the proposal values. The work of Hyari et al. (2016) resembles the most with this research as it is the only research done towards developing an ANN for cost estimation of engineering services. The performance of the model that is described in this research is an improvement with regard to the work proposed by Hyari et al. (2016) as accuracy is higher and deviation of the prediction is lower. Whether the model is an improvement with regard to the pace of completion of tender could not be proven in this research, as no external validation was performed. However, In the interviews, some participant explained that they could provide the information for the variables that were determined for new projects within an hour after reading a RFQ. This would be a major improvement as the current estimation method sometimes takes days or even weeks.

Although the accuracy of the model is relatively high compared to other researches, results from using the model in practice could lack in accuracy. In two test cases, the model predicted the costs of engineering services with an accuracy of 99,7%. In fact, 50% of the testing cases were predicted with an accuracy of 94% or higher. However, in one case the model predicted the costs with accuracy as low as 38%. Furthermore, in two other cases, the model obtained low accuracies of 68,8% and 76% respectively. Eventually, the mean error for the test predictions is 13,65% with a maximum error of 62% and a minimum error of 0,32%. Therefore, while the average accuracy of the testing results is relatively high, the deviation of the individual predictions is still high. In addition, the training of a neural network is involved with two stochastic elements, due to which every training run a different performance will emerge. To get a robust estimate of the skill of a stochastic model, this additional source of variance must be taken into account. model. With the use of the bootstrapping technique, the estimated average MAPE is 61,73% with a standard deviation of 31,27%. Therefore, the more robust estimate of the MAPE of the model is larger compared to the final optimal model. Therefore, while the final model has reasonable accuracy, the model is perceived as very unstable when the additional source of variance due to the stochastic nature of the model is taken into account. Therefore, implementing this method in practice should be considered carefully and is not advised at this moment.

The power of this research lays in the answer to the research question. The research proposed a powerful method that provides guidelines for developing an artificial neural network for cost estimation of engineering services. It provided a solid framework to develop a neural network and improving it. In addition, the method showed how the neural network can be developed into an application to use in practical implementations. The development of the neural network included measures to remove the nuisance from the data and enabled getting only the wisdom from the data that is in it. This method is not only applicable for engineering consultancy firms but can also be used for broader applications of neural network.

6.2 Recommendations and future research

In order to implement an ANN model in an organisation, enough trust should exist towards its capabilities. Therefore, it is recommended to follow the circle of building trust in ANNs as illustrated in Figure 6-1. The developed model should first be tested by means of external validation. This could be done by using the model alongside the currently used detailed estimation method. While carrying out the detailed estimation method, the model can be used to predict the cost of the engineering services. In the end, when the costs are determined by the detailed estimation method, these costs can be compared. By doing so, a feeling towards the practical applicability of the model can be acquired. In addition, when the results of the ANN model are similar to the detailed estimation method, trust is built towards its abilities. This could also imply as a justification for the proposed price for management. Furthermore, the model can function as a cross check for the detailed estimation method. Whenever the prediction of the ANN model is significantly different from the prediction established by the detailed estimation method, there could be made a mistake. This will allow reviewing the process of the detailed estimation method. Therefore, it is recommended to carry out an external validation of the model.



This data can be added to the current database to establish a large sample. So as more data is collected, the neural network gets more accurate by the passing of time. The more data the neural network is developed on, the more variables can be used to explain the problem behind the data and the more accurate the predictions become. Also, by providing more examples, the network can fit the underlying function more accurate as it has more examples to learn from. Therefore, when the dataset is enlarged with a significant amount of data points, the neural network should be redeveloped. Subsequently, an internal validation of the model should be performed to see what the performance of the model is. Thereafter, the application should be rebuilt and an external validation should be

carried out again. If the external validation provides results that are sufficient enough and enough trust is built towards the abilities of the network, it could be implemented in practice. In addition, for implementing the model considering Software as a service (SaaS) can be effective. SaaS is a software distribution model in which a third-party provider hosts applications and makes them available to customers over the Internet. This will allow users within the company to access the application anytime anywhere on any device.

Furthermore, replacing the current detailed estimation method with the ANN model requires additional research. First of all, research needs to be done toward technology adoption. For example, the following question can be asked: what are challenges regarding the adoption of a black box technology within an organization? Neural networks are accurate predictors however, the justification behind the prediction is very hard to do. As the model is developed by means of a self-learning process, it is not known exactly how the relationships between the dependent and independent variables are established. Therefore, bringing out a proposal based only on the ANN still has a lot of challenges. This is an aspect that still needs some further research.

[redacted]

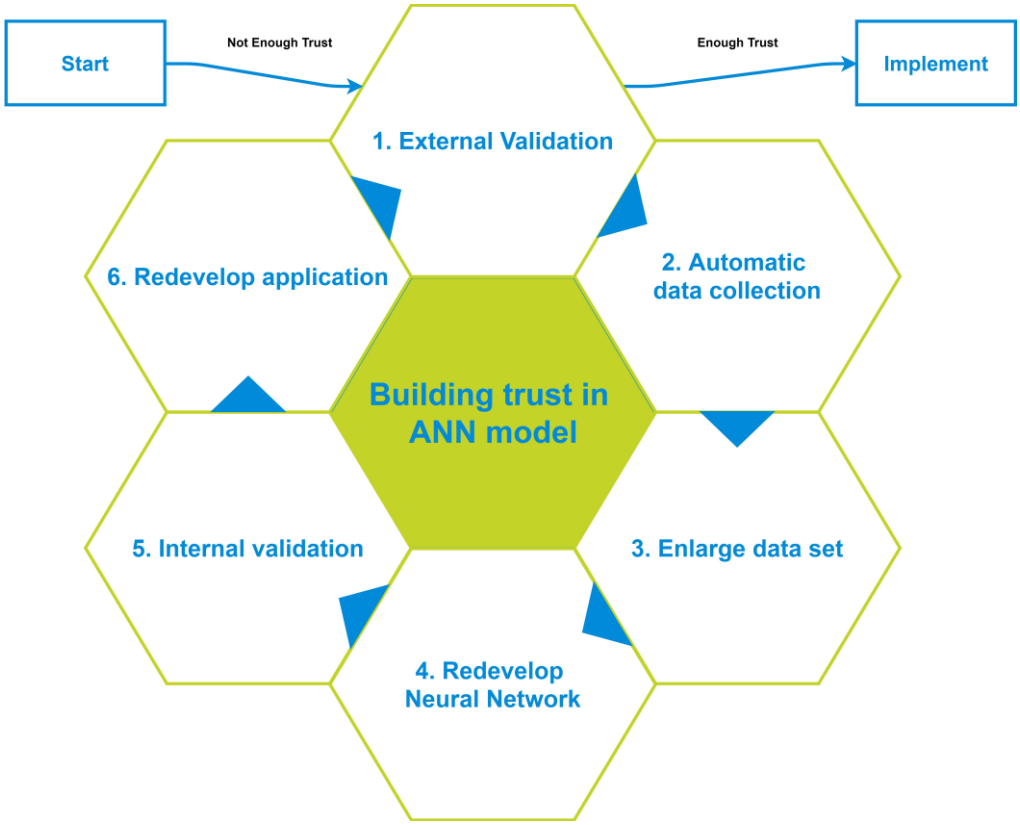


Figure 6-1. Circle of building trust in ANNs

BIBLIOGRAPHY

Ahiaga-Dagbui, D. D., & Smith, S. D. (2012). Neural networks for modelling the final target cost of water projects. *Procs 28th Annual ARCOM Conference*, (September), 307–316. Retrieved from http://hdl.handle.net/1842/6550

Akintoye, A. (2000). Analysis of factors influencing project cost estimating practice. *Construction Management and Economics*, *18*(1), 77–89. https://doi.org/10.1080/014461900370979

Arage, S. S., & Dharwadkar, N. V. (2017). Cost estimation of civil construction projects using machine learning paradigm. *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 594–599. https://doi.org/10.1109/I-SMAC.2017.8058249

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2018). Neural Network Toolbox™; User's guide.

Bosscha, E. (2016). Big data in railway operations : Using artificial neural networks to predict train delay propagation.

Burke, R. (2009). Project Management - Planning and Control Techniques, 251–266.

Busoniu, L., Babuska, R., De Schutter, B., & Ernst, D. (2010). *Reinforcement learning and dynamic programming using function approximators*. CRC press.

Cheng, M. Y., Tsai, H. C., & Sudjono, E. (2010). Conceptual cost estimates using evolutionary fuzzy hybrid neural network for projects in construction industry. *Expert Systems with Applications*, *37*(6), 4224–4231. https://doi.org/10.1016/j.eswa.2009.11.080

Chou, J. S., Yang, I. T., & Chong, W. K. (2009). Probabilistic simulation for developing likelihood distribution of engineering project cost. *Automation in Construction*, *18*(5), 570–577. https://doi.org/10.1016/j.autcon.2008.12.001

Collins, G. S., Groot, J. A. De, Dutton, S., Omar, O., Shanyinde, M., Tajar, A., … Altman, D. G. (2014). External validation of multivariable prediction models : a systematic review of methodological conduct and reporting, 1–11.

Elfaki, A. O., Alatawi, S., & Abushandi, E. (2014). Using intelligent techniques in construction project cost estimation: 10-Year survey. *Advances in Civil Engineering*, *2014*(December). https://doi.org/10.1155/2014/107926

Elkjaer, M. (2000). Stochastic budget simulation. *International Journal of Project Management*, *18*(2), 139–147. https://doi.org/10.1016/S0263-7863(98)00078-7

Emsley, M. W., Lowe, D. J., Duff, A. R. O. Y., & Hickson, A. (2002). Data modelling and the application of a neural network approach to the prediction of total construction costs, 465–472. https://doi.org/10.1080/01446190210151050

Enashassi, A., Mohamed, S., & Abdel-Hadi, M. (2013). Factors Affecting the Accuracy of Pre-Tender Cost Estimates in the Gaza Strip. *Journal of Construction in Developing Countries*.

Engelmore, R. S., & Feigenbaum, E. (1993). Expert systems and artificial intelligence. *Expert Systems*.

Gao. (2009). GAO Cost Estimating and Assessment Guide. *US Government Accountability Office*, (March), 440. Retrieved from http://www.gao.gov/new.items/d093sp.pdf

Günaydin, H. M., & Doğan, S. Z. (2004). A neural network approach for early cost estimation of structural systems of buildings. *International Journal of Project Management*, *22*(7), 595–602. https://doi.org/10.1016/j.ijproman.2004.04.002

Hagan, Howard, Demuth, & Beale. (2014). Neural Network Design (2nd edition).

Hair, J. F. ., Black, W. C. ., Babin, B. J., & Anderson, R. E. (2014). *Multivariate Data Analysis*.

Hamaker, J. (1995). No Parametric estimating. Cost Estimator's Reference Manual. *RD Stewart, RM Wyskida, and JD Johannes, Eds., John Wiley & Sons, New York, NY.*

Hyari, K. H., Al-Daraiseh, A., & El-Mashaleh, M. (2016). Conceptual Cost Estimation Model for Engineering Services in Public Construction Projects. *Journal of Management in Engineering*, *32*(1), 04015021. https://doi.org/10.1061/(ASCE)ME.1943-5479.0000381

Ibrahim, O. M. (2013). A comparison of methods for assessing the relative importance of input variables in artificial neural networks, *9*(11), 5692–5700.

Janssen, J. (2018). Developing a model for estimating road capacity values of weaving sections, 1–90.

Ki, Ö., & Uncuo, E. (2005). Comparison of three back-propagation training algorithms for two case studies, *12*(October), 434–442.

Kim, G. H., An, S. H., & Kang, K. I. (2004). Comparison of construction cost estimating models based on regression analysis, neural networks, and case-based reasoning. *Building and Environment*, *39*(10), 1235–1242. https://doi.org/10.1016/j.buildenv.2004.02.013

Kim, H. J., Seo, Y. C., & Hyun, C. T. (2012). A hybrid conceptual cost estimating model for large building projects. *Automation in Construction*, *25*, 72–81. https://doi.org/10.1016/j.autcon.2012.04.006

Kim, K. J., & Kim, K. (2010). Preliminary Cost Estimation Model Using Case-Based Reasoning and Genetic Algorithms. *Journal of Computing in Civil Engineering*, *24*(6), 499–505. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000054

Kim, P. (2017). *MATLAB Deep Learning*. https://doi.org/10.1007/978-1-4842-2845-6

Kirkham, R. (2014). *Ferry and brandon's cost planning of buildings*. John Wiley & Sons.

Kwak, Y. H., & Watson, R. J. (2005). Conceptual estimating tool for technology-driven projects: Exploring parametric estimating technique. *Technovation*, *25*(12), 1430–1436. https://doi.org/10.1016/j.technovation.2004.10.007

Lebaron, B., & Weigend, A. S. (1995). Evaluating Neural Network Predictors by Bootstrapping, 1–8.

Lester, E. I. A. (2017). Estimating. *Project Management, Planning and Control*, 61–65. https://doi.org/10.1016/B978-0-08-102020-3.00013-9

Mahamid, I. (2013). Conceptual Cost Estimate of Road Construction Projects in Saudi Arabia, *7*(3), 285–294.

Mathworks. (2019). App Building R 2019 a.

Merrow, E. W., S.W., C., & Worthing, C. (1979). A review of cost estimation in new technologies. RAND.

Mirjalili, S. (2018). *Evolutionary Algorithms and Neural Networks*. *Soft Computing and Intelligent Systems: Theory and Applications*. https://doi.org/10.1016/B978-0-12-646490-0.50009-3

Mohammed Arafa and Mamoun Alqedra. (2011). Early Stage Cost Estimation of Buildings Construction Projects using Artificial Neural Networks. Journal of Artificial Intelligence.

Nanculef, R., & Salas, R. (2004). Robust Bootstrapping Neural Networks, (August 2018). https://doi.org/10.1007/978-3-540-24694-7

NASA Executive Cost Analysis Steering Group. (2015). NASA Cost Estimating Handbook. *Nasa*, (4), 63. https://doi.org/10.1017/CBO9781107415324.004

Olden, J. D., & Jackson, D. A. (2002). Illuminating the '" black box "': a randomization approach for understanding variable contributions in artificial neural networks, *154*, 135–150.

Paluszek, M., & Thomas, S. (2017). *MATLAB Machine Learning*. https://doi.org/10.1007/978-1-4842-2250-8

Petroutsatou, K., Georgopoulos, E., Lambropoulos, S., & Pantouvakis, J. P. (2012). Early Cost Estimating of Road Tunnel Construction Using Neural Networks. *Asce*, *138*(June), 679–687. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000479.

Rafiq, M. Y., Bugmann, G., & Easterbrook, D. J. (2001). Neural network design for engineering applications. *Computers and Structures*, *79*(17), 1541–1552. https://doi.org/10.1016/S0045-7949(01)00039-6

Riedmiller, M., & Braun, H. (1992). RPROP - A Fast Adaptive Learning Algorithm, (September 1988).

Son, H., Kim, C., & Kim, C. (2012). Hybrid principal component analysis and support vector machine model for predicting the cost performance of commercial building projects using pre-project

planning variables. *Automation in Construction*, *27*, 60–66. https://doi.org/10.1016/j.autcon.2012.05.013

Steyerberg, E. W., & Harrell, F. E. (2016). Prediction models need appropriate internal , internal e external , and external validation, *69*, 2016.

Tripathi, K. P. (2011). A Review on Knowledge-based Expert System : Concept and Architecture. *Artificial Intelligence Techniques - Novel Approaches & Practical Applications*, *4*(4), 19–23. https://doi.org/10.5120/2845-226

van der Steen, E. (2018). Assessing the influence of tender and project characteristics on project performance.

Zwaving, J. (2014). Probabilistic Estimating of engineering costs. *Delft University of Technology, Delft, The Netherlands*.

APPENDIX A

A.1 Setup survey

Home » Company » Areas & Countries » North West Europe » Project Information Database

Finish Cancel

Please select Project ID number (The Project ID is provided in the table that is sent in the email)

What is the main market type the project was carried out in? *

What type of activity was carried out in the project? *

- ○ Masterplan
- ○ Conceptuel design
- ○ Basic design
- ○ Detailed design
- ○ Basic + Detailed design
- ◉ Specify your own value:

What was the scope of the activities that were carried out? *

- ○ Engineering (E)
- ○ Engineering Procurement Construction management (EPCm)
- ○ Engineering Procurement Construction (EPC
- ◉ Specify your own value:

Was the project a green-field, brown-field project or a combination of green-field and brown-field? *

What is the estimated total investment of the project (Capital Expenditure or Capex)? (Please provide an estimate in the best way possible) *

What was the level of experience of the project manager of Bilfinger Tebodin? *

What was the expected duration of the project in weeks? *

What was the expected number of team members? (The number of all team members is required, not only the core team members) *

What was the status of completeness of the design package before the start of the project? *

	To a small extent		To a moderate extent		To a very great extent
	1	2	3	4	5
Mark one oval	○	○	○	○	○

Was the scope of the work clearly defined in the RFQ and proposal? *

Very poor scope definition	Moderate scope definition	Very good scope definition

	1	2	3	4	5
Mark one oval	○	○	○	○	○

What was the attitude of the client towards design changes? *

	Very high level of cooperation towards approving DCNs		Avarage level of cooperation towards approving DCNs		Very low level of cooperation towards approving DCNs
	1	2	3	4	5
Mark one oval	○	○	○	○	○

What was the level of experience of the team on the client side? *

	Very low level of experience		Moderate level of experience		Very high level of experience
	1	2	3	4	5
Mark one oval	○	○	○	○	○

How demanding was the client with regard to requirements on information and documentation? *

	Very low demands		Standard demands		Very high demands
	1	2	3	4	5
Mark one oval	○	○	○	○	○

Finish Cancel

APPENDIX B

B.1 Input variables quantification

[redacted]

[redacted]			[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]	[redacted]

APPENDIX C

C.1 Results first iterative process

Table C-2. Results training Levenberg-Marquardt backpropagation with 16 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
16-20-1	8.54+04	1.45+11	8.91+10	3.54+10	1.00	0.85	0.83	0.97	0.19	348.76	410.14	115.12
16-19-1	1.28+10	1.76+11	8.17+11	1.59+11	0.99	0.91	0.62	0.88	371.19	946.45	148.70	424.64
16-18-1	2.11+05	2.67+11	1.66+11	6.55+10	0.99	0.78	0.87	0.94	0.17	407.48	191.71	90.91
16-17-1	2.06+05	4.51+11	1.24+11	8.70+10	1.00	0.29	0.61	0.92	0.16	319.87	82.57	61.09
16-16-1	1.23+10	1.10+11	2.72+11	6.65+10	0.97	0.58	0.93	0.94	257.16	1144.26	59.00	361.55
16-15-1	9.16+08	3.36+11	2.30+11	8.64+10	1.00	0.90	0.74	0.92	86.08	250.10	182.63	125.56
16-14-1	2.06+10	1.22+10	1.03+11	3.19+10	0.99	0.94	0.84	0.98	126.16	194.61	124.38	136.26
16-13-1	5.08+10	9.55+10	1.37+11	7.05+10	0.97	0.55	0.73	0.94	158.16	174.48	168.16	162.14
16-12-1	4.48+10	1.03+11	1.29+12	2.42+11	0.94	0.87	0.57	0.76	513.01	959.11	145.44	524.91
16-11-1	1.87+09	2.52+11	6.90+10	5.00+10	1.00	0.65	0.94	0.96	34.92	320.21	87.03	86.04
16-10-1	1.11+09	3.74+10	1.19+12	1.87+11	1.00	0.90	0.77	0.82	58.32	641.88	139.99	159.12
16-9-1	4.86+09	1.51+10	6.66+10	1.58+10	1.00	0.98	0.78	0.99	95.63	199.65	157.79	120.81
16-8-1	1.46+09	5.61+10	9.04+09	1.09+10	1.00	0.88	0.97	0.99	63.15	525.00	73.77	134.74
16-7-1	5.69+10	2.33+10	1.55+12	2.79+11	0.89	0.88	0.56	0.70	308.44	584.37	135.66	324.07
16-6-1	3.84+08	1.07+11	1.34+11	3.68+10	0.99	0.86	0.92	0.97	57.05	152.04	100.18	77.98
16-5-1	4.51+10	6.75+10	9.19+11	1.81+11	0.90	0.93	0.73	0.82	273.07	408.59	106.24	268.33
16-4-1	3.11+09	3.97+11	6.80+09	6.34+10	1.00	0.92	0.73	0.95	90.84	242.39	78.74	111.96
16-3-1	4.58+10	1.17+11	8.27+11	1.75+11	0.91	0.92	0.84	0.84	101.26	176.58	60.75	106.53
16-2-1	7.98+10	1.50+11	8.58+11	2.08+11	0.96	0.67	0.95	0.84	126.21	121.23	103.95	122.08
16-1-1	1.69+11	8.90+09	4.73+10	1.26+11	0.92	0.99	0.84	0.91	143.96	203.63	93.18	145.30

Table C-3. Results training Bayesian regularization backpropagation with 16 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
16-8-1	2.18+04	NaN	4.09+10	6.20+09	1.00	NaN	0.92	0.99	0.09	NaN	149.18	22.68
16-7-1	1.85+05	NaN	1.27+11	1.93+10	1.00	NaN	0.92	0.98	0.44	NaN	130.88	20.20
16-6-1	2.97+07	NaN	3.78+09	5.98+08	1.00	NaN	0.99	1.00	17.02	NaN	99.93	29.58
16-5-1	1.20+08	NaN	2.74+11	4.16+10	1.00	NaN	0.95	0.97	39.57	NaN	79.92	45.68
16-4-1	9.96+07	NaN	1.58+11	2.41+10	1.00	NaN	0.96	0.98	37.25	NaN	50.36	39.24
16-3-1	7.10+08	NaN	1.71+11	2.65+10	1.00	NaN	0.66	0.98	61.80	NaN	57.14	61.10
16-2-1	4.71+09	NaN	1.44+11	2.59+10	1.00	NaN	0.94	0.98	111.36	NaN	54.16	102.69
16-1-1	2.63+10	NaN	1.60+11	4.65+10	0.98	NaN	0.86	0.96	164.36	NaN	52.13	147.35

Table C-4. Results training Resilient backpropagation with 16 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
16-10-1	5.13+09	1.54+11	7.84+11	1.46+11	1.00	0.24	0.68	0.89	226.28	845.41	282.85	328.66
16-9-1	2.53+10	1.67+11	2.41+11	7.94+10	0.98	0.74	0.62	0.92	408.14	377.84	239.16	377.95
16-8-1	9.51+09	4.11+11	4.36+11	1.35+11	0.99	0.62	0.77	0.89	263.70	1190.21	145.97	386.24
16-7-1	3.45+09	4.09+11	1.52+11	8.73+10	1.00	0.94	0.60	0.92	104.81	298.05	136.26	138.85
16-6-1	2.00+09	4.03+10	8.66+11	1.39+11	1.00	0.92	0.75	0.87	59.48	88.68	89.51	68.46
16-5-1	2.76+10	2.79+11	7.72+11	1.78+11	0.95	0.72	0.76	0.83	213.58	443.08	109.02	232.51
16-4-1	3.90+10	1.61+10	7.53+10	4.10+10	0.97	0.98	0.77	0.96	192.83	182.64	153.76	185.36
16-3-1	2.29+10	5.60+10	1.64+11	4.93+10	0.98	0.70	0.87	0.95	165.31	210.02	56.41	155.58
16-2-1	2.94+11	1.80+11	8.77+10	2.45+11	0.79	0.79	0.66	0.78	66.07	67.67	92.88	70.37
16-1-1	2.23+11	4.87+11	7.01+10	2.40+11	0.83	0.70	0.64	0.79	106.86	167.68	46.53	106.93

C.2 Results second iterative process

Table C-5. Results training BR with 15 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
15-8-1	4.12+06	NaN	2.01+11	3.04+10	1.00	NaN	0.80	0.97	2.39	NaN	121.93	20.50
15-7-1	9.83+06	NaN	3.90+11	5.91+10	1.00	NaN	0.84	0.95	8.98	NaN	110.83	24.41
15-6-1	1.33+07	NaN	1.36+11	2.07+10	1.00	NaN	0.87	0.98	12.45	NaN	73.95	21.77
15-5-1	7.52+07	NaN	1.53+12	2.32+11	1.00	NaN	0.50	0.76	25.23	NaN	68.41	31.78
15-4-1	5.43+08	NaN	3.27+09	9.57+08	1.00	NaN	0.83	1.00	61.25	NaN	67.40	62.18
15-3-1	6.88+08	NaN	1.00+11	1.58+10	1.00	NaN	0.94	0.99	56.00	NaN	46.66	54.58
15-2-1	5.60+09	NaN	1.04+11	2.05+10	1.00	NaN	0.77	0.98	117.56	NaN	76.62	111.35
15-1-1	3.70+10	NaN	4.51+11	9.97+10	0.94	NaN	0.96	0.92	181.29	NaN	59.36	162.81

Table C-6. Results training BR with 14 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
14-8-1	1.91+06	NaN	1.45+11	2.20+10	1.00	NaN	0.97	0.98	2.81	NaN	165.19	27.41
14-7-1	2.20+07	NaN	1.54+11	2.34+10	1.00	NaN	0.88	0.98	16.25	NaN	139.16	34.88
14-6-1	5.17+07	NaN	7.95+10	1.21+10	1.00	NaN	0.75	0.99	23.20	NaN	117.73	37.52
14-5-1	2.51+08	NaN	2.61+11	3.98+10	1.00	NaN	0.78	0.96	40.50	NaN	74.00	45.58
14-4-1	5.36+08	NaN	9.51+10	1.49+10	1.00	NaN	0.93	0.99	62.79	NaN	43.67	59.89
14-3-1	7.94+08	NaN	4.74+10	7.86+09	1.00	NaN	0.90	0.99	65.07	NaN	51.79	63.06
14-2-1	6.24+09	NaN	2.79+11	4.76+10	0.99	NaN	0.75	0.96	110.54	NaN	59.82	102.86
14-1-1	2.66+10	NaN	8.59+11	1.53+11	0.97	NaN	0.62	0.87	131.98	NaN	56.21	120.50

Table C-7. Results training BR with 13 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
13-8-1	1.43+05	NaN	3.19+11	4.83+10	1.00	NaN	0.65	0.96	0.80	NaN	137.91	21.57
13-7-1	4.52+07	NaN	7.67+11	1.16+11	1.00	NaN	0.83	0.89	22.92	NaN	85.93	32.47
13-6-1	4.81+07	NaN	6.37+10	9.69+09	1.00	NaN	0.94	0.99	20.35	NaN	59.63	26.30
13-5-1	1.60+08	NaN	3.37+11	5.11+10	1.00	NaN	0.90	0.95	27.93	NaN	56.44	32.25
13-4-1	3.21+08	NaN	3.05+11	4.65+10	1.00	NaN	0.92	0.96	38.82	NaN	41.79	39.27
13-3-1	1.15+09	NaN	1.03+12	1.57+11	1.00	NaN	0.71	0.85	72.93	NaN	45.47	68.77
13-2-1	6.98+09	NaN	1.18+11	2.37+10	0.99	NaN	0.74	0.98	131.38	NaN	72.87	122.51
13-1-1	2.33+10	NaN	5.02+11	9.59+10	0.97	NaN	0.69	0.91	165.02	NaN	59.67	149.05

Table C-8. Results training BR with 12 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
12-8-1	2.87+07	NaN	1.79+11	2.72+10	1.00	NaN	0.98	0.98	18.58	NaN	70.62	26.47
12-7-1	3.28+07	NaN	3.98+10	6.06+09	1.00	NaN	0.91	0.99	14.35	NaN	77.48	23.91
12-6-1	9.64+07	NaN	3.48+11	5.28+10	1.00	NaN	0.94	0.96	37.15	NaN	58.19	40.33
12-5-1	1.59+08	NaN	1.57+11	2.39+10	1.00	NaN	0.88	0.98	44.86	NaN	35.72	43.47
12-4-1	7.21+08	NaN	6.42+10	1.03+10	1.00	NaN	0.92	0.99	71.41	NaN	45.34	67.46
12-3-1	9.71+08	NaN	2.76+10	5.00+09	1.00	NaN	0.95	1.00	83.21	NaN	47.01	77.73
12-2-1	1.08+10	NaN	6.36+10	1.88+10	0.99	NaN	0.92	0.98	113.49	NaN	60.53	105.47
12-1-1	3.52+10	NaN	2.59+11	6.91+10	0.97	NaN	0.79	0.94	168.73	NaN	38.98	149.07

Table C-9. Results training BR with 11 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
11-8-1	5.03+07	NaN	2.05+11	3.11+10	1.00	NaN	0.75	0.97	20.27	NaN	96.52	31.82
11-7-1	8.24+07	NaN	1.32+11	2.01+10	1.00	NaN	0.92	0.98	28.73	NaN	62.28	33.81
11-6-1	2.32+08	NaN	1.47+11	2.25+10	1.00	NaN	0.96	0.98	57.42	NaN	48.32	56.04
11-5-1	2.76+08	NaN	3.28+10	5.21+09	1.00	NaN	0.98	1.00	42.85	NaN	62.82	45.88
11-4-1	7.52+08	NaN	1.20+10	2.46+09	1.00	NaN	0.99	1.00	66.79	NaN	48.71	64.05
11-3-1	1.45+09	NaN	2.01+10	4.28+09	1.00	NaN	1.00	1.00	94.63	NaN	24.37	83.98
11-2-1	4.19+09	NaN	6.05+11	9.53+10	0.99	NaN	0.93	0.93	98.33	NaN	52.96	91.46
11-1-1	5.27+10	NaN	3.41+11	9.64+10	0.92	NaN	0.96	0.92	167.26	NaN	97.04	156.62

Table C-10. Results training BR with 10 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
10-8-1	5.92+07	NaN	1.44+10	2.24+09	1.00	NaN	0.98	1.00	20.81	NaN	72.32	28.62
10-7-1	1.06+08	NaN	4.66+11	7.07+10	1.00	NaN	0.73	0.93	29.01	NaN	58.41	33.47
10-6-1	4.67+07	NaN	5.01+10	7.62+09	1.00	NaN	0.86	0.99	26.69	NaN	46.24	29.65
10-5-1	2.55+08	NaN	5.96+11	9.06+10	1.00	NaN	0.86	0.94	43.41	NaN	49.48	44.33
10-4-1	2.32+09	NaN	2.59+11	4.13+10	1.00	NaN	0.97	0.97	107.33	NaN	46.74	98.15
10-3-1	1.94+09	NaN	9.45+10	1.60+10	1.00	NaN	0.81	0.99	85.09	NaN	48.97	79.62
10-2-1	1.13+10	NaN	3.24+11	5.87+10	0.99	NaN	0.80	0.95	109.59	NaN	50.37	100.61
10-1-1	5.21+10	NaN	4.03+11	1.05+11	0.92	NaN	0.98	0.91	164.15	NaN	71.07	150.05

Table C-11. Results training BR with 9 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
9-8-1	9.07+07	NaN	3.74+10	5.74+09	1.00	NaN	0.96	0.99	32.39	NaN	44.65	34.25
9-7-1	6.43+07	NaN	1.01+11	1.54+10	1.00	NaN	0.79	0.99	27.09	NaN	44.55	29.73
9-6-1	1.40+08	NaN	1.46+11	2.23+10	1.00	NaN	0.84	0.98	29.63	NaN	47.60	32.35
9-5-1	5.21+08	NaN	1.36+11	2.11+10	1.00	NaN	0.96	0.98	48.27	NaN	51.32	48.73
9-4-1	1.53+09	NaN	1.31+11	2.12+10	1.00	NaN	0.97	0.98	73.34	NaN	44.12	68.91
9-3-1	3.22+09	NaN	1.05+11	1.86+10	1.00	NaN	0.90	0.98	94.77	NaN	43.23	86.96
9-2-1	1.29+10	NaN	1.59+11	3.50+10	0.98	NaN	0.95	0.97	160.32	NaN	44.28	142.74
9-1-1	4.48+10	NaN	3.46+11	9.05+10	0.93	NaN	0.91	0.91	187.64	NaN	58.70	168.10

Table C-12. Results training BR with 8 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
8-8-1	2.52+07	NaN	1.36+11	2.07+10	1.00	NaN	0.98	0.98	17.23	NaN	50.66	22.30
8-7-1	2.92+08	NaN	1.49+10	2.51+09	1.00	NaN	0.97	1.00	49.08	NaN	68.49	52.02
8-6-1	1.42+08	NaN	5.61+10	8.63+09	1.00	NaN	0.94	0.99	29.95	NaN	51.27	33.18
8-5-1	6.36+08	NaN	1.16+10	2.30+09	1.00	NaN	0.96	1.00	55.07	NaN	42.26	53.13
8-4-1	1.20+09	NaN	1.52+11	2.40+10	1.00	NaN	0.85	0.98	57.07	NaN	49.15	55.87
8-3-1	1.11+09	NaN	7.51+11	1.15+11	1.00	NaN	0.97	0.90	67.69	NaN	49.40	64.92
8-2-1	2.48+09	NaN	2.35+10	5.67+09	1.00	NaN	0.98	0.99	97.76	NaN	34.02	88.10
8-1-1	1.44+10	NaN	5.10+10	2.00+10	0.99	NaN	0.83	0.98	138.59	NaN	31.46	122.36

Table C-13. Results training BR with 7 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
7-8-1	1.13+08	NaN	1.84+11	2.80+10	1.00	NaN	0.97	0.98	32.14	NaN	55.44	35.67
7-7-1	1.40+08	NaN	9.18+10	1.40+10	1.00	NaN	0.85	0.99	32.29	NaN	52.16	35.30
7-6-1	1.10+08	NaN	1.52+10	2.40+09	1.00	NaN	0.99	1.00	25.18	NaN	48.82	28.76
7-5-1	7.08+08	NaN	3.06+10	5.24+09	1.00	NaN	0.96	1.00	46.73	NaN	37.41	45.32
7-4-1	8.90+08	NaN	8.16+10	1.31+10	1.00	NaN	0.99	0.99	72.77	NaN	50.39	69.38
7-3-1	2.45+09	NaN	6.27+10	1.16+10	1.00	NaN	0.80	0.99	72.37	NaN	45.75	68.34
7-2-1	4.92+09	NaN	6.91+10	1.46+10	1.00	NaN	0.85	0.99	87.09	NaN	38.55	79.73
7-1-1	1.49+10	NaN	9.96+11	1.64+11	0.98	NaN	0.98	0.94	117.24	NaN	44.00	106.15

Table C-14. Results training BR with 6 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
6-8-1	7.55+07	NaN	2.09+10	3.23+09	1.00	NaN	0.91	1.00	18.55	NaN	46.53	22.79
6-7-1	3.24+08	NaN	9.01+09	1.64+09	1.00	NaN	0.95	1.00	35.19	NaN	32.83	34.83
6-6-1	9.19+08	NaN	8.35+10	1.34+10	1.00	NaN	0.92	0.99	42.86	NaN	37.00	41.97
6-5-1	6.06+08	NaN	2.65+11	4.07+10	1.00	NaN	0.86	0.96	44.06	NaN	40.42	43.51
6-4-1	8.37+08	NaN	1.09+11	1.72+10	1.00	NaN	0.95	0.99	47.59	NaN	39.18	46.31
6-3-1	2.70+09	NaN	3.51+11	5.54+10	0.99	NaN	0.93	0.95	68.97	NaN	27.32	62.66
6-2-1	6.46+09	NaN	2.25+11	3.96+10	0.99	NaN	0.92	0.97	106.80	NaN	46.74	97.70
6-1-1	4.71+10	NaN	1.71+11	6.58+10	0.96	NaN	0.83	0.94	89.39	NaN	34.32	81.05

Table C-15. Results training BR with 5 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
5-8-1	1.64+08	NaN	4.03+11	6.11+10	1.00	NaN	0.77	0.94	34.36	NaN	47.42	36.34
5-7-1	2.77+08	NaN	4.65+11	7.06+10	1.00	NaN	0.88	0.95	40.87	NaN	48.20	41.98
5-6-1	4.79+08	NaN	3.86+09	9.91+08	1.00	NaN	1.00	1.00	33.15	NaN	27.41	32.28
5-5-1	1.52+09	NaN	8.45+09	2.57+09	1.00	NaN	0.98	1.00	66.21	NaN	30.67	60.82
5-4-1	1.69+09	NaN	2.57+11	4.03+10	1.00	NaN	0.87	0.97	65.02	NaN	42.71	61.64
5-3-1	2.66+09	NaN	2.07+10	5.40+09	1.00	NaN	0.96	1.00	83.33	NaN	41.36	76.97
5-2-1	7.52+09	NaN	1.69+11	3.20+10	0.99	NaN	0.93	0.97	54.72	NaN	44.23	53.13
5-1-1	4.52+10	NaN	1.90+11	6.71+10	0.96	NaN	0.71	0.94	95.92	NaN	53.40	89.48

Table C-16. Results training BR with 4 input variables

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
4-8-1	1.13+11	NaN	7.67+11	2.12+11	0.81	NaN	0.78	0.79	186.61	NaN	95.21	172.76
4-7-1	3.38+09	NaN	6.44+10	1.26+10	1.00	NaN	0.78	0.99	86.01	NaN	76.70	84.60
4-6-1	9.23+10	NaN	8.67+11	2.10+11	0.83	NaN	0.92	0.80	110.97	NaN	57.93	102.93
4-5-1	7.69+09	NaN	1.24+11	2.53+10	0.99	NaN	0.94	0.98	112.37	NaN	71.82	106.23
4-4-1	1.58+10	NaN	9.29+10	2.74+10	0.99	NaN	0.95	0.98	120.14	NaN	62.43	111.40
4-3-1	2.64+10	NaN	1.89+11	5.11+10	0.97	NaN	0.94	0.95	131.87	NaN	67.96	122.19
4-2-1	3.42+10	NaN	3.08+11	7.57+10	0.97	NaN	0.85	0.93	106.68	NaN	67.28	100.71
4-1-1	5.89+10	NaN	6.35+11	1.46+11	0.94	NaN	0.73	0.86	112.67	NaN	48.49	102.94

Table C-17. Results training BR with 7 input variables (MLR)

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
7-8-1	9.69+07	NaN	1.29+10	2.62+09	0.9999	NaN	0.9830	0.9976	25.38	NaN	59.73	32.15
7-7-1	2.32+08	NaN	9.79+10	1.95+10	0.9998	NaN	0.7614	0.9821	40.49	NaN	54.45	43.24
7-6-1	3.16+08	NaN	4.20+10	8.53+09	0.9998	NaN	0.8419	0.9921	46.68	NaN	52.70	47.87
7-5-1	4.91+08	NaN	5.88+11	1.16+11	0.9995	NaN	0.6830	0.9014	42.78	NaN	41.19	42.47
7-4-1	2.82+09	NaN	2.79+10	7.76+09	0.9978	NaN	0.8703	0.9928	116.80	NaN	47.24	103.10
7-3-1	3.88+09	NaN	7.22+10	1.73+10	0.9969	NaN	0.8024	0.9840	83.73	NaN	42.07	75.53
7-2-1	1.57+10	NaN	9.59+11	2.01+11	0.9720	NaN	0.6191	0.7936	110.28	NaN	71.17	102.57
7-1-1	6.28+10	NaN	9.93+10	7.00+10	0.9451	NaN	0.8492	0.9330	166.66	NaN	60.53	145.75

Table C-18. Results training BR with 6 input variables (MLR)

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
6-8-1	1.10+08	NaN	2.09+10	4.20+09	0.9999	NaN	0.9698	0.9961	30.95	NaN	38.57	32.45
6-7-1	1.59+08	NaN	1.12+11	2.22+10	0.9999	NaN	0.9024	0.9808	22.60	NaN	41.99	26.42
6-6-1	4.07+08	NaN	3.14+10	6.51+09	0.9996	NaN	0.9784	0.9939	38.83	NaN	42.56	39.56
6-5-1	2.31+09	NaN	4.72+10	1.11+10	0.9980	NaN	0.9712	0.9905	71.34	NaN	52.55	67.64
6-4-1	1.70+09	NaN	9.98+10	2.10+10	0.9986	NaN	0.9419	0.9828	49.46	NaN	58.50	51.24
6-3-1	6.43+09	NaN	1.08+12	2.19+11	0.9911	NaN	0.4325	0.7728	99.58	NaN	46.21	89.07
6-2-1	1.39+10	NaN	1.85+11	4.75+10	0.9750	NaN	0.9373	0.9549	127.62	NaN	44.31	111.21
6-1-1	3.55+10	NaN	6.96+11	1.66+11	0.9383	NaN	0.7517	0.8371	97.45	NaN	59.30	89.94

Table C-19. Results training BR with 5 input variables (MLR)

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
5-8-1	2.69+08	NaN	2.57+10	5.28+09	0.9998	NaN	0.8089	0.9952	34.30	NaN	38.56	35.14
5-7-1	1.27+08	NaN	1.06+11	2.09+10	0.9999	NaN	0.9065	0.9806	23.56	NaN	42.47	27.28
5-6-1	5.69+08	NaN	9.31+10	1.88+10	0.9995	NaN	0.9241	0.9841	57.35	NaN	39.90	53.91
5-5-1	3.10+09	NaN	5.63+10	1.36+10	0.9950	NaN	0.9841	0.9885	66.16	NaN	37.00	60.42
5-4-1	2.74+09	NaN	5.68+10	1.34+10	0.9975	NaN	0.9469	0.9876	74.37	NaN	44.80	68.54
5-3-1	1.51+10	NaN	3.73+10	1.94+10	0.9872	NaN	0.9799	0.9826	81.56	NaN	50.89	75.52
5-2-1	3.73+10	NaN	2.37+10	3.46+10	0.9677	NaN	0.9815	0.9676	96.38	NaN	39.69	85.21
5-1-1	4.55+10	NaN	3.41+11	1.04+11	0.9238	NaN	0.9644	0.9134	180.66	NaN	40.04	152.96

Table C-20. Results training BR with 7 input variables (Expert Opinion)

Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
7-8-1	7.14+09	NaN	5.94+11	1.23+11	0.9888	NaN	0.7691	0.8800	138.35	NaN	108.34	132.44
7-7-1	7.44+10	NaN	1.26+12	3.08+11	0.8440	NaN	0.6564	0.6701	245.33	NaN	100.64	216.83
7-6-1	7.37+09	NaN	2.74+11	5.99+10	0.9933	NaN	0.7892	0.9473	164.45	NaN	161.38	163.84
7-5-1	9.99+10	NaN	1.01+12	2.79+11	0.7407	NaN	0.8243	0.7370	263.20	NaN	149.72	240.85
7-4-1	2.23+10	NaN	2.17+11	6.06+10	0.9814	NaN	0.7698	0.9456	167.52	NaN	142.04	162.50
7-3-1	3.95+10	NaN	7.78+10	4.71+10	0.9656	NaN	0.8876	0.9554	212.98	NaN	118.43	194.36
7-2-1	5.05+10	NaN	5.39+10	5.12+10	0.9583	NaN	0.9449	0.9519	167.69	NaN	121.04	158.50
7-1-1	1.34+11	NaN	7.36+11	2.53+11	0.7396	NaN	0.7708	0.7440	305.55	NaN	146.00	274.13

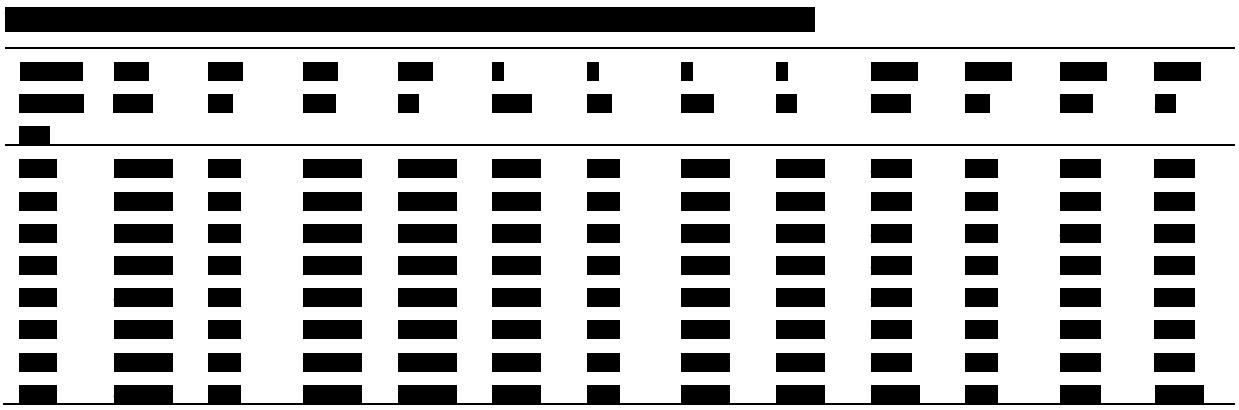
Table C-21. Results training BR with 6 input variables (Expert Opinion)

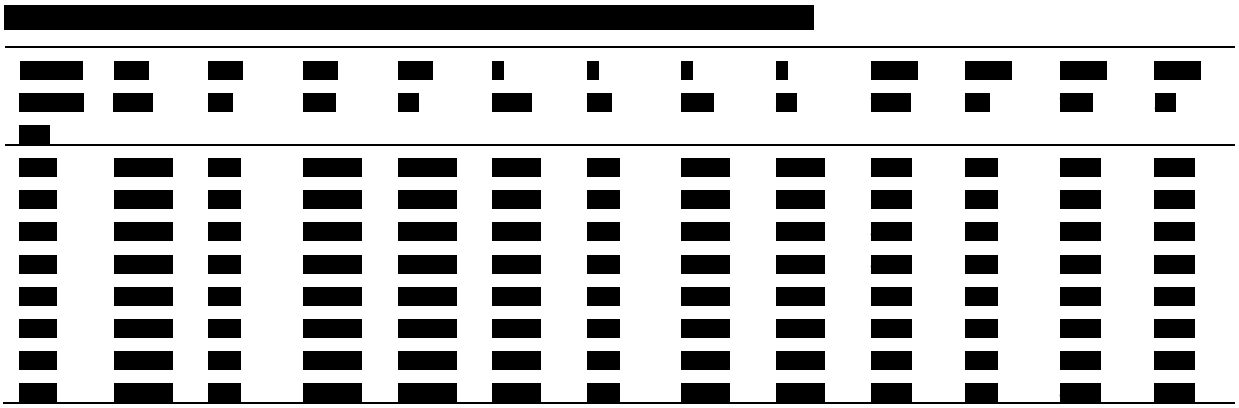
Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
6-8-1	9.75+10	NaN	9.78+11	2.71+11	0.6931	NaN	0.7899	0.7511	303.81	NaN	106.12	264.87
6-7-1	2.06+10	NaN	9.90+10	3.60+10	0.9824	NaN	0.9405	0.9668	210.61	NaN	105.18	189.84
6-6-1	2.88+10	NaN	1.01+12	2.22+11	0.9755	NaN	0.8105	0.8585	232.38	NaN	113.38	208.94
6-5-1	1.77+10	NaN	1.39+12	2.88+11	0.9687	NaN	0.2917	0.6847	142.27	NaN	122.50	138.38
6-4-1	2.25+10	NaN	3.34+11	8.39+10	0.9791	NaN	0.6268	0.9194	270.88	NaN	129.77	243.08
6-3-1	5.08+10	NaN	4.83+11	1.36+11	0.9560	NaN	0.5713	0.8772	254.72	NaN	130.20	230.19
6-2-1	1.25+11	NaN	7.20+11	2.42+11	0.8049	NaN	0.6627	0.7428	276.58	NaN	139.18	249.51
6-1-1	1.16+11	NaN	8.96+11	2.69+11	0.7609	NaN	0.7238	0.7237	304.73	NaN	134.29	271.16

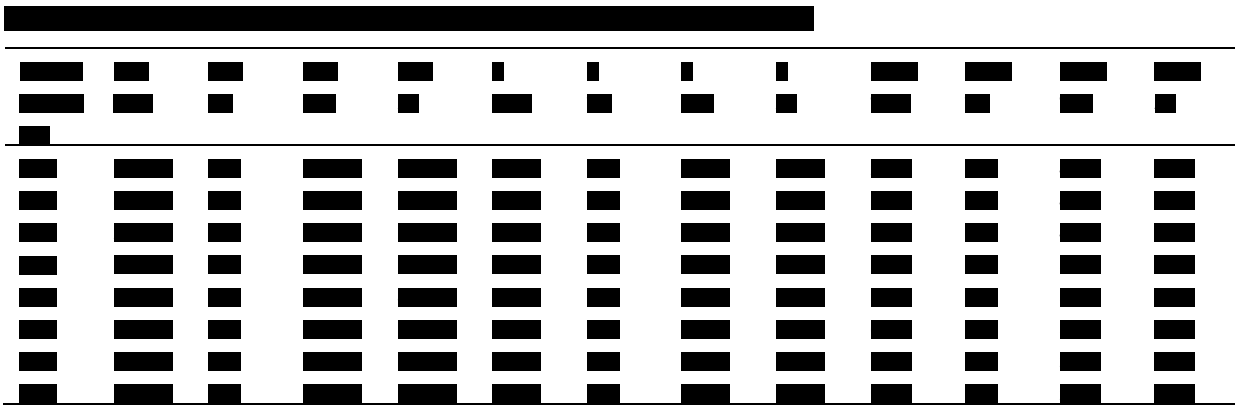
Table C-22. Results training BR with 5 input variables (Expert Opinion)

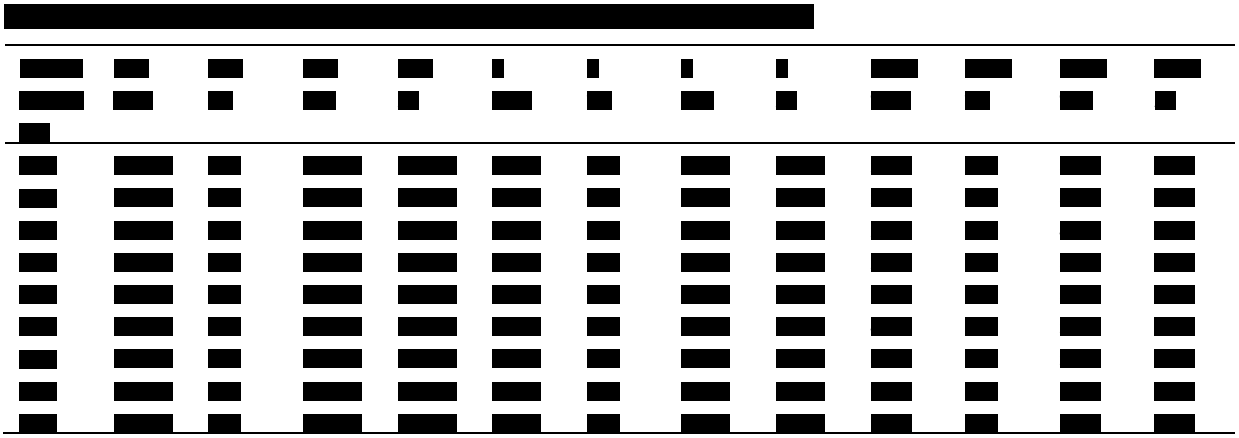
Network Architecture	MSE Train	MSE Val	MSE Test	MSE All	R Train	R Val	R Test	R All	MAPE Train	MAPE Val	MAPE Test	MAPE All
5-8-1	2.55+11	NaN	8.25+10	2.21+11	0.7569	NaN	0.8539	0.7681	294.68	NaN	94.56	255.27
5-7-1	2.18+11	NaN	2.39+11	2.22+11	0.7609	NaN	0.7774	0.7663	302.21	NaN	105.39	263.44
5-6-1	2.17+11	NaN	2.65+11	2.27+11	0.7530	NaN	0.8609	0.7664	274.98	NaN	93.25	239.19
5-5-1	1.19+11	NaN	1.08+12	3.08+11	0.6923	NaN	0.7270	0.6858	327.93	NaN	131.40	289.22
5-4-1	1.23+11	NaN	8.20+11	2.60+11	0.7706	NaN	0.6675	0.7218	314.34	NaN	110.55	274.20
5-3-1	2.27+11	NaN	2.15+11	2.24+11	0.7566	NaN	0.9375	0.7692	259.57	NaN	117.41	231.57
5-2-1	2.51+11	NaN	9.82+10	2.21+11	0.7796	NaN	0.7708	0.7685	276.79	NaN	128.65	247.61
5-1-1	1.28+11	NaN	8.26+11	2.66+11	0.7683	NaN	0.8405	0.7319	252.00	NaN	98.60	221.78

[redacted]



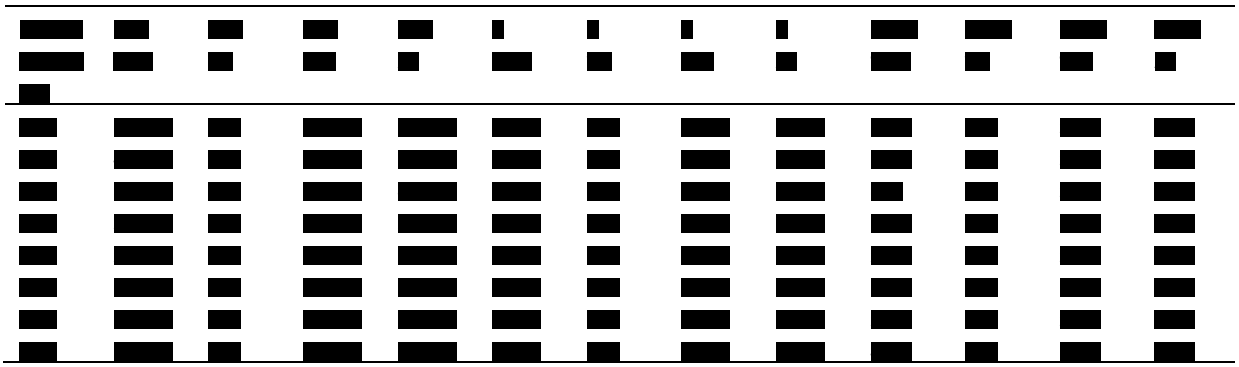




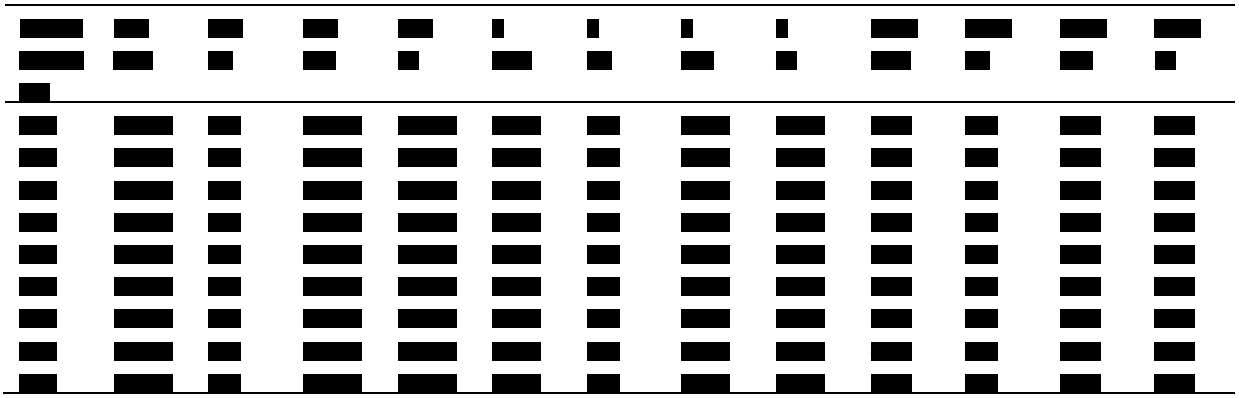


[redacted]

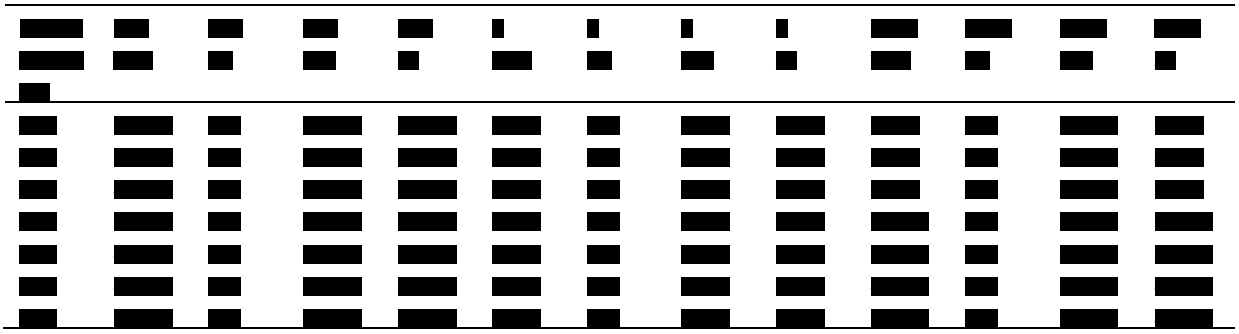
[redacted]



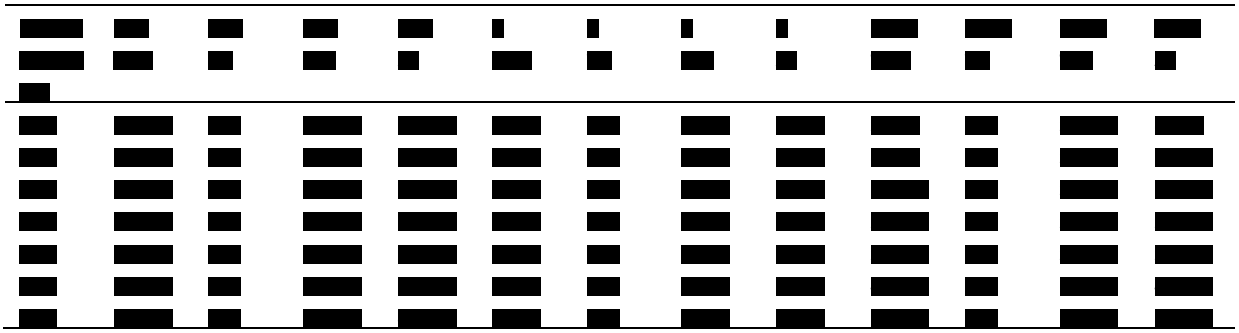
[redacted]



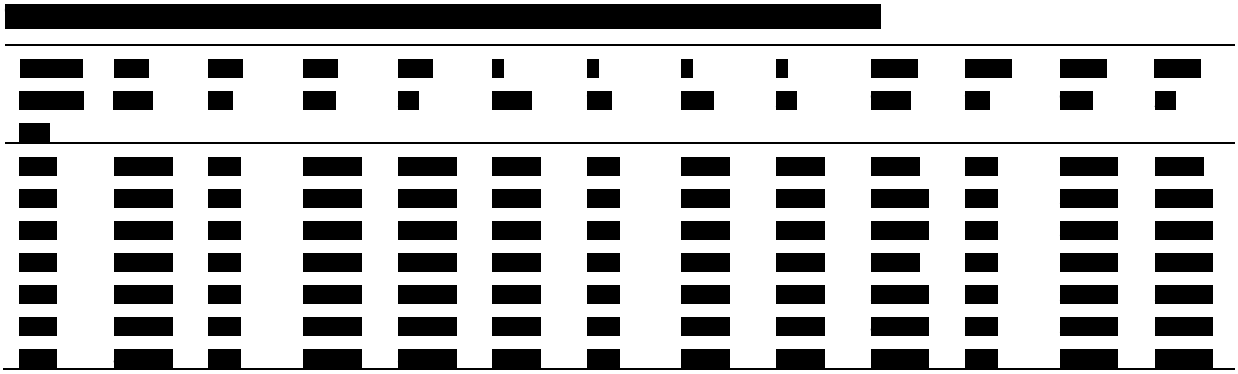
[redacted]



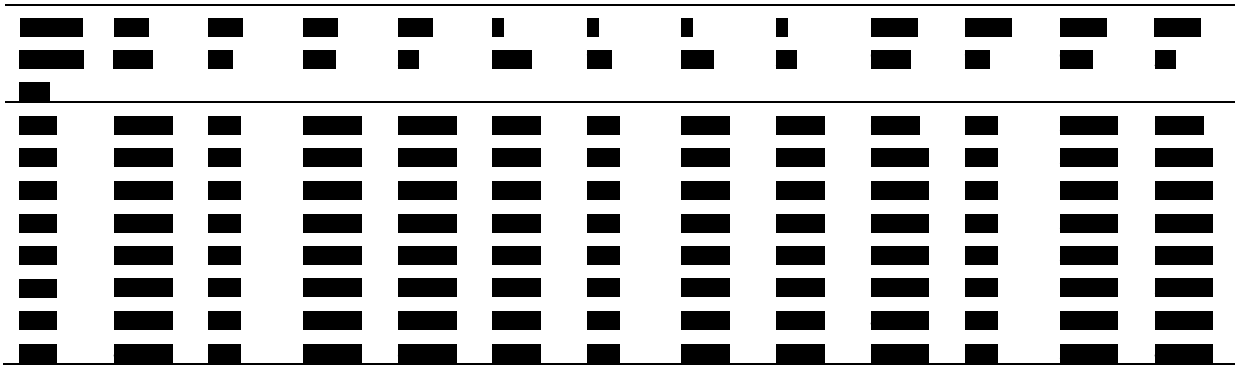
[redacted]



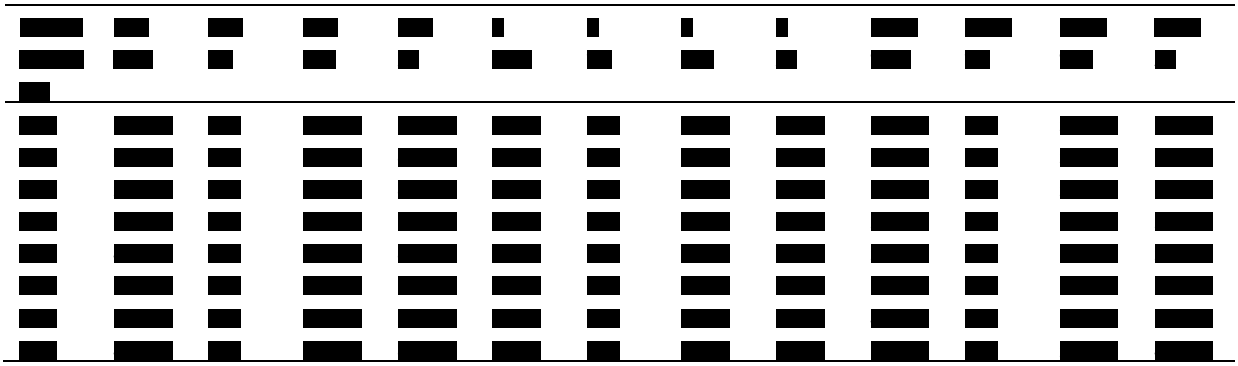
[redacted]



[redacted]



[redacted]



APPENDIX D

D.1 Average MAPE and standard deviation of models 'multistart'

Table D-38. Performance of neural networks of 'multistart'

Model nr.	Mean absolute percentage error
1	52.7462749
2	79.10468938
3	59.96292058
4	53.20109772
5	96.18556908
6	92.86942347
7	27.53216057
8	51.94879465
9	24.49296377
10	142.7362528
11	76.77861914
12	67.5389498
13	37.74131488
14	80.09244664
15	106.3743214
16	73.88741515
17	29.66288738
18	49.20169994
19	38.21399939
20	36.22354276
21	25.49078788
22	102.0901055
23	22.98482827
24	60.07005323
25	74.18151233
26	89.01768891
27	95.30796837
28	157.4454966
29	55.85660251
30	111.400547
31	34.20472838
32	81.8341454
33	28.06606196
34	52.42700518
35	89.18845721
36	13.64807251
37	107.2680416
38	46.84501062
39	29.56033487
40	42.20521074
41	82.46891567
42	41.82120113
43	38.77114275
44	21.46755582
45	42.7656826
46	23.34691306
47	58.461616
48	79.77529201
49	25.90044004

Model nr.	Mean absolute percentage error
50	115.096704
51	83.81848271
52	30.97732489
53	67.91739119
54	43.06860621
55	86.87614675
56	82.34114273
57	95.1973282
58	80.02821517
59	79.82711992
60	32.10749783
61	81.20049019
62	45.13213193
63	94.48862895
64	24.27262438
65	54.92150316
66	78.38710582
67	43.81154352
68	35.4603349
69	39.56977774
70	97.75732149
71	59.84128692
72	59.11561302
73	38.74918668
74	104.7778755
75	60.11330598
76	60.56390918
77	53.30747646
78	59.36396556
79	70.00244925
80	49.79608155
81	107.1632282
82	70.6693832
83	37.51948871
84	18.57977544
85	36.47277337
86	36.88870184
87	48.62294861
88	31.65860803
89	50.88117885
90	140.2928271
91	38.5077569
92	93.58833989
93	153.5507925
94	30.828782
95	35.70483823
96	28.8169582
97	27.57291089
98	34.33203804
99	79.87698141
100	53.37755008
Mean	61,73%
Standard deviation	31,27%