DATA INTEGRITY AND CLASSIFICATION IN NUCLEAR POWER PLANT DATA MANAGEMENT

Lappeenranta–Lahti University of Technology LUT

Master's Degree Programme in Data Analytics in Decision Making

2022

Joona Mussalo

Examiners: Associate Professor, Kalle Elfvengren

Post-Doctoral Researcher, Antti Ylä-Kujala

ABSTRACT

Lappeenranta–Lahti University of Technology LUT

School of Engineering Science

Industrial Engineering and Management

Joona Mussalo

Data integrity and classification in nuclear power plant data management

Master's Thesis

2022

57 pages, 7 figures, 10 tables and 1 appendix

Examiners: Associate Professor Kalle Elfvengren and

Post-Doctoral Researcher Antti Ylä-Kujala

Keywords: data quality, data integrity, classification

Data integrity is the cohesion of the data in the context of its use. The definition of integrity differs between sources, but in this case, it describes the fitness for use in the nuclear power plant project. It can be measured by using data quality dimensions, quality, accuracy, completeness, and timeliness. In complex engineering projects, data integrity is critical for the successful execution of the project. Issues in integrity will harm the traceability and the management of data. These issues can be caused by poor implementation of the data management system, data procedures, or lack of cooperation between involved parties.

This study aims to identify the issues in a data management system and its processes, which could cause data integrity issues. The study discusses the concepts of data quality and classification. A literature review was carried out to observe the potential classification methods and pre-processing techniques. The classification model was built using Weka, which is open-source software for machine learning and data mining. It contains a great variety of algorithms for machine learning and tools for data pre-processing. The J48 classification algorithm was chosen to build a classifier for data imputation.

The results of this study suggest that a data imputation task can be performed for the reference data in the system. However, more algorithms should be tested to optimize the classification accuracy. The experiment was done on one data set. For future work, classification should be extended to other data that is found in the system. The identified integrity issues were mainly related to the problem of multiple systems that are used simultaneously. The reference data is obtained from multiple sources which are not integrated. The data objects and relationships may have been changed in one system and not in the other. The issues with incorrect or missing reference data are escalated when new design documents are received, and the relationships cannot be established.

TIIVISTELMÄ

Lappeenrannan–Lahden teknillinen yliopisto LUT

School of Engineering Science

Tuotantotalous

Joona Mussalo

Datan yhtenäisyys ja luokittelu ydinvoimalan tiedonhallintajärjestelmässä

Tuotantotalouden diplomityö

2022

57 sivua, 7 kuvaa, 10 taulukkoa ja 1 liite

Tarkastajat: Dosentti Kalle Elfvengren ja tutkijatohtori Antti Ylä-Kujala

Avainsanat: datan laatu, datan yhtenäisyys, luokittelu

Datan eheydellä tarkoitetaan sen yhtenäisyyttä käyttökohteessa. Datan eheyden määritelmä vaihtelee lähteiden välillä, mutta tässä tapauksessa se kuvaa käyttökelpoisuutta ydinvoimalaitosprojektissa. Sitä voidaan mitata datan laadun ulottuvuuksien avulla, keskittymällä tarkkuuteen, ajantasaisuuteen, täydellisyyteen ja laatuun. Monimutkaisissa rakennusprojekteissa datan eheys on ratkaisevan tärkeää projektin onnistumisen kannalta. Ongelmat eheydessä haittaavat datan jäljitettävyyttä ja hallintaa. Nämä ongelmat voivat johtua datanhallintajärjestelmän huonosta toteutuksesta, tietomenettelyistä tai osapuolten välisen yhteistyön puutteesta.

Tämän tutkimuksen tavoitteena on tunnistaa datanhallintajärjestelmän ja sen prosessien ongelmat, jotka voivat aiheuttaa datan eheysongelmia. Tutkimus käsittelee datan laadun ja luokituksen käsitteitä. Kirjallisuuskatsaus suoritettiin mahdollisten luokittelumenetelmien ja esikäsittelytekniikoiden tarkkailemiseksi. Luokittelumalli rakennettiin Wekalla, joka on avoimen lähdekoodin ohjelmisto koneoppimiseen ja tiedon louhintaan. Se sisältää laajan valikoiman algoritmeja koneoppimiseen ja työkaluja tietojen esikäsittelyyn. J48-luokittelualgoritmi valittiin luokittelijan rakentamiseen puuttuvan datan päättelyä varten.

Tutkimuksen tulokset viittaavat siihen, että puuttuvia viitedatan arvoja voidaan päätellä muiden arvojen avulla. Luokittelutarkkuuden optimoimiseksi tulisi kuitenkin testata lisää algoritmeja. Tutkimuksessa käytettiin yhtä datajoukkoa. Tulevaisuuden työskentelyä varten luokittelu tulisi laajentaa koskemaan myös muuta järjestelmästä löytyvää dataa. Tunnistetut eheysongelmat liittyivät pääasiassa useiden samanaikaisesti käytettävien järjestelmien ongelmaan. Viitedata tuodaan järjestelmään useasta lähteestä, joita ei ole integroitu. Viitedatan arvoja on saatettu päivittää yhdessä järjestelmässä, mutta toisessa on vielä vanhaa tietoa. Virheelliseen tai puuttuvaan viitedataan liittyvät ongelmat eskaloituvat, kun uusia suunnitteludokumentteja vastaanotetaan, eikä yhteyksiä datan välillä voida muodostaa.

ACKNOWLEDGEMENTS

I would like to thank Marko Juslin, Arne Monsholm and Jouni Raila for supporting and providing assistance for my master’s thesis. I would also like to express gratitude to my co-workers and the company, which provided me with this interesting topic to work with.

Tampere, 16.6.2022

Joona Mussalo

ABBREVIATIONS

ADLAS	Advanced Nuclear Licensing and Safety Design Method for Nuclear Facilities
EPC	Engineering Procurement and Construction
GBS	Geographical Breakdown Structure
ID3	Iterative Dichotomiser 3
IEC	International Electrotechnical Commission
IRT	Integrated Review Tool
ISO	International Organization for Standardization
kNN	The K Nearest Neighbour
MAR	Missing At Random
MCAR	Missing Completely At Random
MNAR	Missing Not At Random
PBS	Plant Breakdown Structure
PSAR	Preliminary Safety Analysis Report
SDV	Smart Data Validator
SPF	SmartPlant Foundation
SSC	Structures, Systems and Components
UNECE	United Nations Economic Commission for Europe
WBS	Work Breakdown Structure
Weka	Waikato Environment for Knowledge Analysis
YVL	Ydinturvallisuusohjeet
DHP	Data Handover Package

Table of contents

Appendices

LIST OF FIGURES

LIST OF TABLES

1 Introduction

The first chapter defines the research topic and gives an overview of the subject. The background for the study is presented, objectives are described, and research questions are defined. This chapter also presents the research method used in the thesis and the structure of this thesis.

1.1 Background

The integrity of the data is in important role in complex engineering projects, where configuration management approaches are used over the project's lifecycle. Configuration management is a process of maintaining system integrity while controlling the changes made to both the dataset and the real-world engineering system it describes (Lindkvist et al., 2013). It has become a more and more important process in the era of Big Data and is commonly used in areas such as aerospace engineering, civil engineering, and power plants.

Data integrity is needed for traceability and tracking during the project. If continuous tracking is lost, it can cause problems. During the project, there will be situations, where historical information is needed for upcoming changes. Creating these changes will be a lot easier task if the backtracking process is not too long. Historical information is also needed during the project to inspect previous decisions to make sure everything is done according to the plan.

Faults and errors in the data will affect the review process, making it cost more resources and time. This makes the review process more inefficient and can cause delays in the project. Errors in the data can lead to wrong decisions made in the process, which are extremely hazardous in a complex engineering project, that has safety as its priority.

Transfer of the data is a crucial part of maintaining the data integrity. After the data transfer and validation process, a comparison between the initial data and the new data in the target system should be performed. This ensures that the loading process doesn't corrupt the data. The validation process also must be well defined and procedures for missing and invalid data must be in place.

1.2 Objective and scope

The objective of this study is to find, what can cause problems to data integrity and what effects it does have on the review process. The study gives the reader an overview of data integrity and explains its importance to configuration management. The use of classification methods is reviewed for missing data imputation. Previous literature is used to examine research done on the topic.

The research questions for this topic are:

1. *How are the problems in data integrity affecting the review process?*
2. *What are the causes and impacts of poor quality of data?*
3. *How machine learning can be used to improve data?*

The main focus of this study is on the data that is used in the review tool, which is part of the plant information management system. The plant information management system was implemented in the project during 2021 when a system with the ability to support the plant design structure was needed. This system will be used as the main plant information management system during the project and operating phase of the power plant. During the planning phase, the main focus is on the review of the safety and usability of the plant.

The causes of poor quality of the data in the IRT are analysed and their effects on the review process will be evaluated. After the data has been analysed and conclusions about the quality of the data have been made, a method for improving the current data is proposed. Classification method is used to create a classifier to predict missing data values. This study is limited to the data in the review tool. That will give a good overall view of the project's data cohesion.

1.3 Research methods

The research methods of this thesis are literature review and case study. A literature review is used to study previous work done in the areas of data integrity, configuration management, data validation, and classification. The case study presents a case project, which gives the reader a better understanding of the topic at hand. The search words, search scope and research databases used are presented in Table 1. The search words contain topics related to data quality, data integrity and machine learning.

Literature databases and search engines that are used for the literature review are Google Scholar, ScienceDirect, Ebsco, and SpringerLink. Scientific articles and books are the main sources for the theoretical part of this study. Some online articles are also reviewed regarding the topic.

Table 1. Search words, scope and databases.

Search words	**Search Scope**	**Research Databases**
• Data Integrity • Data Quality • Data Transfer • Data Validation • Classification	• Power Plants • Energy Production	• Google Scholar • Science Direct • Ebsco • SpringerLink

1.4 Structure

This thesis is divided into seven chapters. The first chapter is an introduction, which defines the objective and presents the background for this study. The second chapter is about data integrity and data quality. The third chapter explains how the classification algorithms can be used for data imputation. The fourth chapter explains the research methodology. The fifth chapter introduces the case project for this thesis and describes the data used in the study.

Chapter six is about the classification of the reference data within the system. Chapter seven discusses the conclusions about the topic, and answers to research questions.

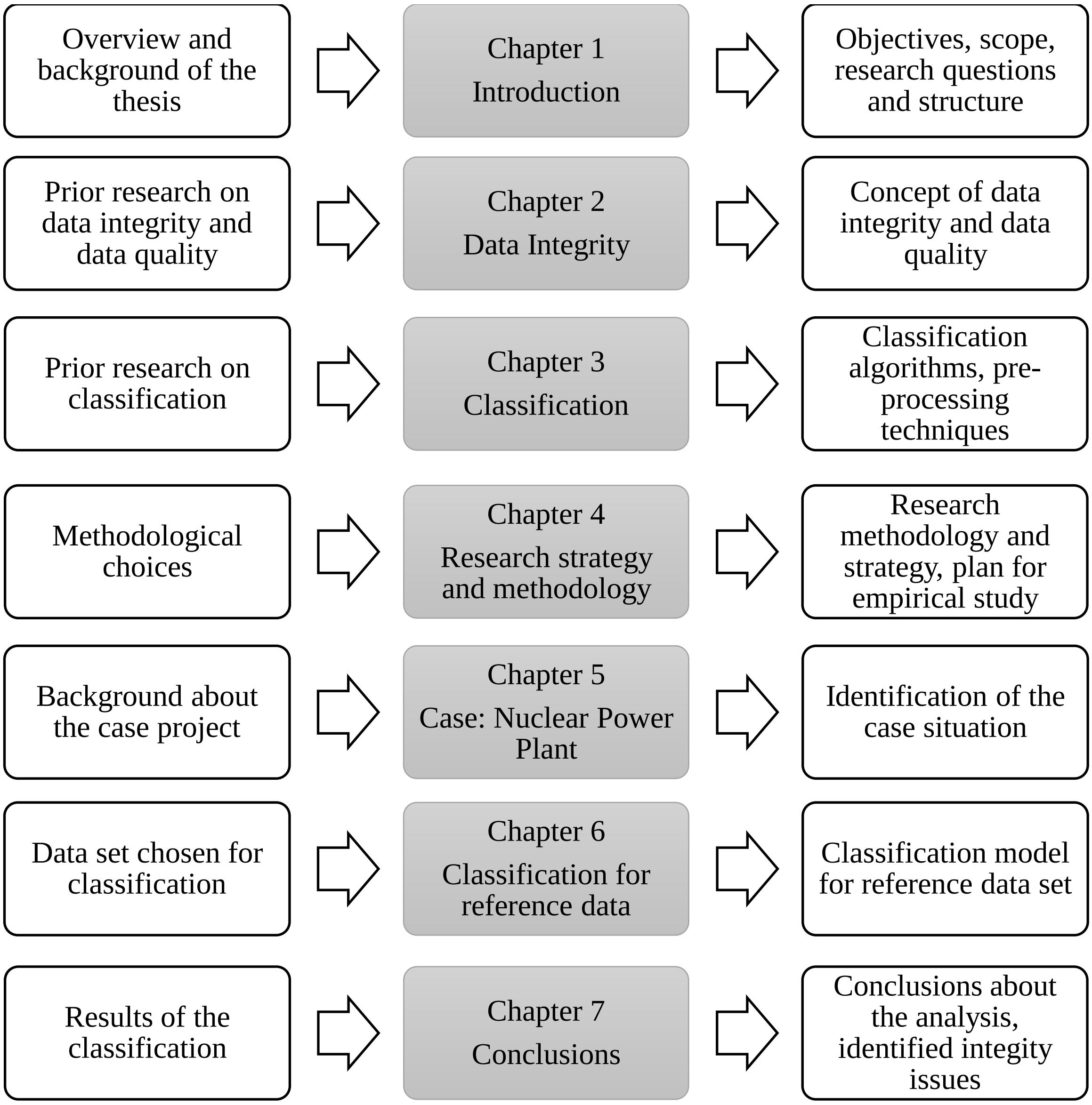


Figure 1. Input-output chart.

Figure 1. explains the inputs and outputs of each chapter. The empirical part of the study starts from chapter 5.

2 Data Integrity

This chapter defines the concept of data integrity and data quality. They are related concepts and data quality can be seen as a part of data integrity, which is the main topic of this thesis.

2.1 Definition

Data integrity is in an essential role when talking about managing a complex engineering project. In projects, where regulatory agencies are involved, integrity becomes an even more significant factor. Regulators are examining the data during inspections, and the correctness of the information cannot be verified, if it's inconsistent or not following the set guidelines. In a nuclear power plant project, the regulatory inspection is a key part of the process. The inspection is done to check if the design of the plant ensures safe operation and is following the rules set for the nuclear industry.

Integrity is based on the ability to answer several questions (Lagoze, 2014). Lagoze (2014) defines these questions as:

1. *What is the origin of these data?*
2. *Who has been responsible for them since their origination?*
3. *Can we apply our standard notions for trust and integrity to them?*
4. *Do our standard methodologies for interpreting them and drawing conclusions from them make sense?*

With the help of these questions, the integrity of the data can be defined on a conceptual level, using the notions of "trust", fitness for use" and "consensual understanding" (Lagoze, 2014). It tells us that the data used is accurate and consistent and is a reliable source for decision-making in the project. On the other hand, if the integrity is lost, the data cannot be trusted, and incorrect choices will be made.

2.2 Ensuring the data integrity

Ensuring the data integrity should be done by identifying the possible scenarios and steps in the data process that may cause integrity loss. The most common reasons for data issues are human errors, transfer errors and software issues. Human errors refer to situations, where data is unavailable because it was accidentally deleted, or the data entry has typing errors. Transfer errors can happen when the data file is corrupted, or the validation rules are incomplete. (Sharma, 2021) The data structure differences between systems can also cause an issue because some relations and references can be lost during the transfer.

The validation of data is the most important step in avoiding data quality issues. UNECE (2013) defines validation as: “An activity aimed at verifying whether the value of a data item comes from the given (finite or infinite) set of acceptable values.” The set of acceptable values may be set as allowed values for a single field. It may also refer to a set of valid value combinations for a record, column, or larger collection of data (Bosch et al., 2016). If the validation passes, it will not guarantee that the data is correct, but if it doesn’t pass, the data is incorrect. Typical checks that are performed on data entries are range checks, type checks, length checks, and format checks. A range check ensures that the value is between an upper and lower bound. A type check ensures that the value is desired type, for example, date or integer. Length checks can be used if the number of characters can be defined. A format check is commonly used to check if dates are presented correctly. The purpose of a validation process is to identify data that is incorrect or missing and fix those issues before using the data for operations.

Validation can be performed on many levels. The levels can be seen as a reflection of the stages in the production process (Simon, 2013). These validation levels are described in Table 2 based on Simon’s (2013) study. The most common practice is to use the level 0 and level 1 validations when loading data to the database.

Table 2. Validation levels.

Validation levels	Description
Level 0	The structure of the file is checking the number of columns, the format of columns, and missing data check.
Level 1	Intra-dataset check. Values in specified columns are compared to the list of acceptable values.
Level 2	The content of the file is compared with the content of other files referring to the same domain. "Intra-domain, intra-source checks"
Level 3	The content of the file is compared with the content of other files referring to the same domain. Mirror checks verify the consistency between declarations from different sources referring to the same phenomenon.
Level 4	Consistency checks between separate domains available in the same institution.
Level 5	Consistency checks between the data available in the institution and the data available outside the institution.

2.3 Quality of the data

The quality of the data is the main part of data integrity. In the ISO/IEC 25012 standard for data quality model, the quality is defined as the "degree to which the characteristics of data satisfy stated and implied needs when used under specified conditions" (ISO/IEC, 2008). The quality can be divided into many dimensions, depending on the context where the data is used. A data quality dimension is a characteristic or a part of the information for classifying information and requirements of the data (Affendey et al., 2012). The number of dimensions presented varies between studies. Affendey et al. (2012) introduce 40 dimensions for quality, whereas Boone et al. (2014) use the more common approach, in which the data quality is divided into four dimensions. Those are accuracy, timeliness, consistency, and completeness. These dimensions are presented as the most important in many studies, including Boone et al. (2014) and Blake and Mangiameli (2011). Examples and descriptions of these dimensions are presented in Table 3.

Table 3. Common quality dimensions. (Boone et al., 2014)

Data quality dimension	**Description**	**Example**
Accuracy	Is the data free of errors?	The Data values are matching between different systems
Consistency	Is the data presented in the same format?	The same date format is used in all data instances
Timeliness	Is the data up to date?	The data is checked and updated after a defined period
Completeness	Is necessary data missing?	All attributes have values in a data set

The cost of poor-quality data is estimated by some data quality experts as being from 15% to 25% of the company’s operating profit. In a recent survey of 599 companies, poor data management is costing global businesses more than 1.3 billion euros per year in billing, accounting, and inventory snafus alone. Most of that cost is related to the accuracy dimension. (Olson, 2003)

Lee et al. (2002) suggest that one set of metrics is not sufficient to evaluate data quality in all situations. There are subjective and objective data quality metrics that can be used as a guideline when assessing the quality of the data set.

Accuracy

Accuracy refers to the degree to which the data values are corresponding to “real” values (Boone et al., 2014). This dimension can be evaluated by comparing the values to the “real” values that are considered correct. Loshin (2009) states that the accuracy is quite challenging to monitor because the real-world information may change over time and corroborative data might not be available as reference data. If corroborative data is available as reference data, an automated process can be used to verify the accuracy, otherwise, a manual process is needed.

Consistency

Batini and Scannapieco (2009) suggest that consistency can be divided into two categories of integrity constraints. Intra-relation constraints define the range of acceptable values for the attribute. Inter-relation constraint states that equal values should be found from both relations for each entity. Consistency can be defined to an extent in which data are always presented in the same format and are compatible with previous data (Strong et al., 2013.)

Timeliness

Timeliness as a data quality dimension refers to the time expectation for accessibility and availability of information (Loshin, 2009). It can be measured as the time between when the information is expected and when it’s available for use. Even if the information accuracy and consistency are high, the data quality can still be considered as low, if it takes a lot of time to get the data in usable form to the database. A target date should be set for data loading, which defines the allowed delay between data arrival and readiness for use.

Completeness

Completeness can be defined as “the extent to which data are of sufficient breadth, depth, and scope for the task at hand” (Batini and Scannapieco 2016). Depending on the context where the data is used, the completeness can vary. In two different scenarios, the same data set can be valued as complete or incomplete, based on the complexity of the task.

Missing data

Missing data is defined as the data value that is not stored for a variable in the observation of interest (Kang, 2013). Missing values are relatively common in all data sets, and they cannot be completely avoided. The missing data can be divided into three types, based on the reasons why they are missing (Graham, 2009). These types are missing completely at random (MCAR), missing at random (MAR), and missing not at random (MNAR). Kang (2013) describes these different types of missing data as follows. MCAR is defined as when the probability that the data are missing is not related to the specific value which is supposed to be obtained or the set of observed responses. In the case of MAR, the missing data is dependent on the observed responses but is not related to specific values. The last type, MNAR, means that there is some undiscovered factor that is causing missing values, and it cannot be defined. These different types of missing data have statistical definitions, but often their practical meaning is elusive (Graham, 2009). However, recognizing these types of missing data from the data set can help to identify the ways of fixing these issues.

According to Kang (2003), the most common approach to the missing data is to simply ignore those cases where missing data is present and analyse the remaining data. This approach works well if the values are missing completely at random. Other types of missing data can cause biased results if they are simply ignored.

Considering a single source of information at the instance level, missing data is a quite common problem. It can correspond to an empty value, or to a placeholder value indicating that there is no data. This issue affects the completeness dimension since data is not present. (Bernardino et al., 2015)

Duplicate data

The original aim of data cleansing was to eliminate duplicates from a data set (Freytag and Müller, 2005). Duplicate data mainly causes two types of errors. The first one is incorrect or incomplete references to other data tables. One of the entities might have correct relations and up-to-date values, while the other one has not. If the wrong entity is used, it will cause issues. The other issue with duplicates is related to actions that are done for each item in a list. For example, a letter will be sent to every person on a list, which leads to some people receiving multiple letters. This leads to unnecessary expenses and frustration (Freytag and Müller, 2005).

Incorrect data

Incorrect data occurs when a value does not match up to the real entity. That data is then considered inaccurate. A misspelling is a common cause of incorrect data. (Bernardino et al., 2015) Incorrect data can exist when multiple systems or databases are used and the data is not updated simultaneously in both locations.

2.4 Causes for poor quality of data

The data quality is often compromised by many factors, including data entry errors, missing integrity constraints, and multiple conventions for recording information. Data entry errors happen when there are spelling mistakes. The “same” value can be accidentally entered multiple times, and some relations may be incorrect because of that. Integrity issues are caused when there is no validation for the accepted value range, for example, “year”=20200. Multiple conventions issue refers to values, which are entered in different formats. The company name can be, for example, “Company” or “Company Ltd”. (Elmagarmid, 2007) Many of these errors can be avoided with a proper validation process, where data entries are checked before loading them to the database.

Structural heterogeneity occurs when the fields of tuples are structured differently in different databases (Elmagarmid, 2007). The same information can be stored in one field in one database and multiple fields in the other database.

Data that is accurate when initially created, can become inaccurate when time passes. The data value is not changing, but its accuracy does. This applies to many types of information. One of the most common causes of accuracy decay is personal information in an employee database. Most employees do not update their information when something changes and many times the information is only accurate when the person joins the company. (Olson, 2003, p.50). Other commonly decaying instances are objects that are assigned to some employee. That employee may leave the company or change the role within the company, which causes the information to be inaccurate.

3 Classification

Machine learning algorithms can be divided into three main types based on the data that is used for learning. Supervised learning, unsupervised learning, and reinforcement learning. The most widely used machine learning methods are supervised learning methods (Mitchell and Jordan, 2015). In Supervised learning, the data is labelled. The model is trained until it can detect relationships and patterns between the data and its labels. Unsupervised learning does not use labelled data. It uses unlabelled data for analysing and finding patterns without a human interception. This chapter focuses on classification, which is a supervised learning method.

3.1 Introduction to classification

Supervised classification is one of the most frequently carried out machine learning tasks. The goal of supervised learning is to build a model of the distribution of class labels in terms of predictor features. The resulting classifier is then used to assign class labels to the testing instances where the values of the predictive features are known, but the value of the class label is unknown (Kotsiantis et al., 2007). The classification is useful in cases where one attribute of the data instance is missing but can be predicted from other attribute values. The simplest scenarios are binary classification problems where the label class has two possible values. One of the use cases is the classification of spam emails. Based on a certain set of attributes, the email can be classified as “spam” or “real”. The class label can be a continuous or a discrete value. If the output takes discrete values that indicate the class label, the learned mapping leads to the classification of the input data. If the output takes a set of continuous values, it leads to a regression of the input (Wu and Liu, 2012).

The classification accuracy and meaningfulness are dependent on the distribution of the class attribute. If a high percentage of the output values belong to one class, high classification accuracy can be achieved by simply always choosing that class. The most informative and reliable results are achieved when the class attribute values are distributed evenly.

3.2 Classification algorithms

The purpose of the classification algorithm is to construct a classifier and then analyse the characteristics of the unknown data. The performance of the classifier is measured by its accuracy to predict the class label values. Many different algorithms can be used for classification, including decision tree methods, instance-based learning methods, artificial neural network methods, and the Bayesian method. (Gao et al., 2009) The fitness of an algorithm for a specific task is dependent on many factors. For example, a rule-based decision tree performs efficiently when the data is globally well distributed, and an instance-based algorithm can take good advantage of the data's local characteristics. (Ali et al., 2017) Usually, the best way of finding a suitable classification algorithm for the given data set is to try different algorithm types and compare the results. Some of the popular algorithms are presented in the next paragraph.

The K Nearest Neighbour (kNN) method is widely used in the applications of data mining and machine learning due to its simple implementation and eminent performance. The main idea of the kNN method is to predict the label of a test data point by the majority rule. The label is predicted with k most similar training data points in the feature space. The kNN classification has at least two open issues to be addressed. Those are the similarity measurement between two data points and the selection of the k value. (Cheng et al., 2017) Various distance functions have been used in the literature to measure the similarity. Hu et al. (2016) compare Euclidean distance, Chi-square, and Minkowsky distance for categorical and numerical data sets. All those functions seem to perform somewhat equally for categorical data. Using the Chi-square leads to better results in datasets, where the number of attributes is greater. For numerical data, there is no significant difference in performance. The best distance algorithm changes between the datasets, but each of them can be used without fear of decreased accuracy. In mixed datasets, where categorical and numerical data are present the Chi-square is again the best performer. From an overall perspective, it seems that the Chi-square is the best choice for all types of datasets.

The Chi-square distance between two samples, x and y, is calculated as:

$$\chi^2 = \frac{1}{2}\sum_{i=1}^{n}\frac{(x_i - y_i)^2}{(x_i + y_i)}$$

The Euclidean distance for high dimensions is calculated as:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

where p and q are points in a multi-dimensional space.

Choosing the *k*-value for the kNN algorithm has been a favoured research topic in data mining and machine learning since the algorithm was proposed. Traditional kNN classification methods select the *k* value by either setting a fixed constant for all test data or conducting cross-validation to estimate the k value for each test data point (Cheng et al., 2017). The simplest method for choosing the *k* value is to take a square root of the sample size. This gives reasonably accurate results but usually is not the most optimal value. More accurate results can be achieved by performing the classification multiple times using values from k to N and choosing the value where the classification accuracy is maximum. Despite this method finding the most optimal value, it can be time and resource-intensive, if the sample size is large. Cheng et al. (2017) have proposed a method for solving the *k*-value by computing the correlations between the training data points and choosing the number of correlated data points as the *k* value. Ali et al. (2017) are suggesting that analysing the dataset characteristics with the help of historic datasets can yield values that are in close proximity to the optimal *k* values.

The C4.5, also known as J48 for its Java implementation, is a decision tree-based algorithm commonly used for supervised classification tasks. C4.5 is a successor of the ID3 algorithm, and it was developed by Ross Quinlan in 1993 (Quinlan, 1993). It has been taken as a point of reference for the development and analysis of novel proposals. The results show that the C4.5 algorithm provides good classification accuracy and is one of the fastest among the compared machine learning and data mining algorithms (Ruggieri, 2002). The C4.5 was included in the top 10 data mining algorithms identified by the IEEE International Conference on Data Mining in 2006 (Kumar et al., 2007). The algorithm can be used with nominal and numerical attributes, and it uses information gain as the basis of splitting attribute selection. When continuous attributes are used for creating the classifier, they need to be discretized first. The algorithm will discretize those attributes and calculate information gain for all splitting points. This will lead to a significant increase in operation time when selecting attributes. Another flaw of the algorithm is that it generates as many branches for each node as there are values in the class attribute. This leads to empty branches which yield no contribution to the classification and can cause overfitting. (Gao and Wang, 2021) With feature selection, the operation time can be reduced greatly.

The J48 algorithm in Weka uses Information Gain as a decisive measurement for splitting the nodes. It compares the splitting scenarios to find out the optimal splitting attributes by calculating the entropy of child nodes.

The Information Gain is calculated as a reduction in Entropy after splitting the parent node S using the attribute A as:

$$IG(S,A) = E(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} E(S_v)$$

where $E(S)$ is the entropy of the parent node before the split and $E(S_v)$ is the entropy of the child node after the split.

The entropy in this case means the impurity of the node. The entropy E is calculated as:

$$E = \sum_i -p_i \log_2 p_i$$

where p_i is the probability of class i.

The entropy of the child nodes should be as small as possible, which is achieved when the node only contains instances belonging to one class. That results in high information gain value, which corresponds to high classification accuracy.

3.3 Data pre-processing

Data pre-processing is the first step that should be done when performing any machine learning operations on a data set. There is a hierarchy of problems that are often encountered in data preparation:

1. *Impossible values have been inputted.*
2. *Unlikely values have been inputted.*
3. *No values have been inputted (missing values).*

4. *Irrelevant input features are present in the data at hand. (Kotsiantis et al., 2007)*

Impossible values are generally straightforward, such as negative prices or measurements when positive ones are expected. These can be corrected or just simply removed and handled as missing data (Kotsiantis et al., 2007).

Unlikely values, also known as outliers or anomalies, should be identified from the data to get more accurate results. Outlier detection is an extensive research branch in data mining due to its use in a wide range of applications. By identifying outliers, researchers can obtain vital knowledge which assists in making better decisions about data. Detecting outliers translates to significant actionable information in a large variety of applications such as fraud detection. An Outlier is generally considered a data point that significantly differs from other data points, or which does not match to the expected pattern of the phenomenon it describes. (Bah et al. 2019) Outlier or anomaly detection is related to noise removal. Noise can be defined as a phenomenon in data that is not interesting to the analyst. A distinction between anomalies and noise is that anomalies can be interesting for analysis, but noise should be always removed because it acts as a hindrance to data analysis. (Banerjee et al., 2009) Outliers and anomalies are simpler to identify from numerical attributes, where distance-based and density-based approaches can be utilized. For categorical attributes, the context of the data must be known for being able to define which value is extraordinary.

The simplest way of dealing with missing values is to discard the instances that contain them. This method is only practical when the data contains a relatively small number of examples with missing values. (Garcia et al., 2014) When instances with missing values take up a large proportion of the data, removing them would reduce the amount of data too much, which can cause inaccuracies in the classification model. Juhola and Laurikkala (2011) estimated the number of missing values needed to noticeably harm the prediction accuracy in classification. The classifications were performed with nearest neighbour searching when the number of missing values was between 0 to 30%. For the two-class dataset, the accuracy didn’t differ much between 0 and 30%. However, when there are more than two classes, more than 20% of missing values seem to cause inaccuracies. The more classes of different sizes there are, the more sensitive the data set becomes. If the class distribution is somewhat equal, the number of missing values can be higher without reducing the accuracy noticeably.

3.4 Feature selection

Feature subset selection is the process of identifying and removing as many irrelevant attributes as possible. This reduces the dimensionality of the data and enables classification algorithms to operate faster and more efficiently. (Kotsiantis et al., 2014) The goal of attribute subset selection is to find a minimum set of attributes such that the accuracy of the model is as close as possible to the original accuracy with all attributes. It also simplifies the patterns that are discovered, making them easier to understand.

There are various methods for finding the most suitable attributes for the classification model. These methods can be divided into three categories: filter methods, wrapper methods and embedded methods. Filter methods use variable ranking techniques as the criteria for variable selection. Ranking methods are used due to their simplicity and good performance for practical applications. Ranking criteria is used to score the variables, and those below the defined threshold are removed. These methods are filter methods since they are applied before the classification itself. (Chandrashekar, 2013) Popular methods to rank attributes are calculating their correlation to the class attribute and calculating their Information gain regards to the class.

Correlation as a filter method can be used in two ways to select features. In the first method, each of the attributes are individually compared to the class attribute by calculating the correlation between them. Then some arbitrary threshold is set and attributes that have correlation below that threshold are removed. If the data includes nominal attributes, they are considered on a value-by-value basis by treating each value as an indicator (Weka, SourceForge). Correlation values range from -1 to 1. Values close to 1 indicate a strong positive correlation and values close to -1 indicate a strong negative correlation.

The correlation between a and b is calculated as:

$$\mathrm{r}_{ab} = \frac{cov\ (a,b)}{\sqrt{var(a)} * \sqrt{var}(b)}$$

where $cov(a,b)$ is the covariance, and $var(a)$ and $var(b)$ are the variances of a and b.

The correlation can be also calculated between the attributes. If two variables are highly correlated, it's unnecessary to keep both for model building. Removing one of them simplifies the model while the accuracy doesn't change much.

The Information gain can also be used for feature selection. In the context of feature selection, the Information gain is also known as Mutual Information (MI). It is a measure between two variables that calculates the amount of information that one variable has about the other variable (Estevez and Vergara, 2015).

The Mutual Information is defined as:

$$MI(x;y) = \sum_{i=1}^{n}\sum_{j=1}^{n} p(x(i),y(j)) \log(\frac{p\big(x(i),y(j)\big)}{p\big(x(i)\big)p\big(y(j)\big)})$$

where MI is zero when x and y are statistically independent. (Estevez and Vergara, 2015)

Wrapper methods consider feature subsets by the performance of a modelling algorithm. The performance evaluation is repeated for each subset. The subset generation is dependent on the search strategy that is used. Wrappers are a lot slower than filters in finding sufficiently good subsets because the classification model must be evaluated for each subset. However, they usually give better results than filter methods for the exact reason. (Bogunovic et al., 2015) Wrappers can be further divided into two groups, sequential selection algorithms and heuristic search algorithms. Sequential selection algorithms are iterative algorithms. The Sequential Feature Selection (SFS) algorithm starts with an empty set and adds one feature for the first step which gives the highest value for the model. In the next step, the remaining features are added individually to the current subset and the new subset is evaluated. The best of those subsets is chosen for the next round, where the process repeats. This continues until the model performance doesn't increase anymore or the desired number of features is chosen. A Sequential Backward Selection (SBS) algorithm can also be used. It is similar to the SFS, but now all features are present at the start. The algorithm removes the lowest performing attributes until the set threshold is met. (Chandrashekar, 2013)

Embedded feature selection methods are popular because they perform feature selection while constructing the classifying model, removing the need for pre-processing feature selection step. Some popular algorithms are Least Absolute Shrinkage and Selection Operator (LASSO) and Random Forests. (Holm and Mangal, 2018)

When choosing a feature selection method, the characteristics of the data set should be taken into a consideration. Some methods perform better with numerical data and some with categorical data. The size of the data set also matters, for large data sets some methods are not efficient since they are computationally expensive. The focus is on the filter methods because they are the easiest to apply and usually perform well with all types of data sets.

Figure 2. shows common filter methods and when they should be used. The Input variable refers to the attributes in the data set and the Output variable is the class.

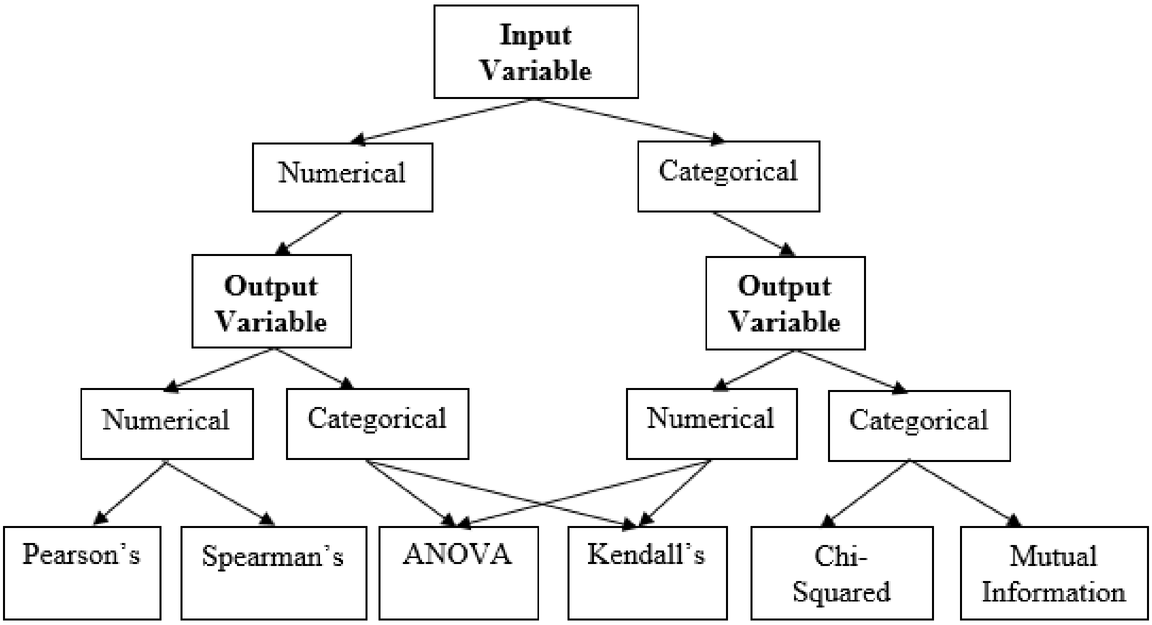


Figure 2. Choosing filter method based on variables.

In many cases the dataset has input variables that are not the same data type. There can be numerical and categorical data in the same data set, which makes choosing the feature selection method more complicated. In this situation where it's unclear which method will perform the best, a good solution is to try multiple selection algorithms and compare the results. Usually, the results are not too far away from each other and a subset of attributes for the model building can be compiled from those results. Even if there are some guidelines for choosing the right method, each dataset is unique, and the optimal method may not be found.

3.5 Overfitting and cross-validation

Ying (2019) states that overfitting is a fundamental issue in supervised machine learning which prevents us from perfectly generalizing the models to fit observed data on training data, as well as unseen data on the testing set. Overfitting happens because of the presence of noise, the limited size of the training set and the complexity of classifiers. (Ying, 2019) The feature selection helps to reduce the model's complexity, which also decreases the chance of overfitting the model. Hardt et al. (2019) conducted a meta-analysis of overfitting in machine learning and based on their results, there is surprisingly little evidence of substantial overfitting. However, when the holdout method is used, the danger of overfitting raises. The holdout is an approach where training data and test data are separate and cross-validation is not used.

Liu (2016) explains cross-validation as a statistical method for evaluating and comparing learning algorithms by dividing data into two segments. One is used to train the model and the other is used to validate the model. The training set and validation sets must cross over in successive rounds such that each data instance is being validated against. The basic form of cross-validation is k-fold cross-validation, where the k-value is the number of equally sized segments (Liu et al., 2016). If the k-value is 10, there will be 10 validation rounds, where a different segment of the data is used for training each time. The other nine segments are used for validation. 10-fold cross-validation is commonly used and since the classifiers are developed with 90% of the full data, the variance between different seeds is generally low (Simon, 2007).

4 Research strategy and methodology

In this thesis, multi-stage research is used to find the answers to research questions. Common stages for research projects are clarifying the topic, reviewing the literature, designing the research, collecting data, analysing data and writing up (Lewis et al., 2015). These stages are utilized in this thesis, but the process is not so straightforward. The stages are most likely revisited more than once since new findings are discovered during the research process and some adjustments will be done.

A case study is chosen as a research strategy for this thesis. There are two basic types of designs that are possible when using case studies for explanatory purposes. The first type is a single-case design. In a single-case design, the study focuses on one instance, which is used to describe the topic. The second type is a multiple-case design, where conclusions are drawn from a group of cases. (Yin, 1981) A single-case design is selected to fit this topic best, where data integrity is evaluated in a specific project. It helps the reader to understand the concept of data integrity with a concrete example from a complex real-world engineering project.

The empirical part of this thesis explains the data process in the project. The critical points for possible data corruption can be then recognized from the process. When those have been identified, the causes and solutions for data corruption can be found. After, the current data is analysed for integrity.

5 Case: Nuclear Power Plant

This chapter introduces the case project for this thesis. First, it describes the background of the project, followed by the empirical study.

5.1 Background

In the empirical case study, the data process in the nuclear power plant design phase is reviewed. The goal is to identify any points, where data integrity may be compromised and find solutions to prevent those issues. After that, the data in the plant data management system is analysed to find out if there are some integrity issues. In the end, a classification model is built for the imputation of the missing data.

At the beginning of the project, the design review and data transfer from the plant supplier were done on a document basis. M-Files was used as a document management system, but its limitations as a complete tool for plant data management were noticed. A system that can handle grouping by the plant structure and attributes of the physical equipment was needed. In 2021, the Hexagon SDx was deployed as a comprehensive system for managing plant data. At the same time, a new format for data transfer between the power company and the plant supplier was developed.

The SDx system will be the main plant information management system during the planning and operating phase of the plant. During the planning phase, the main focus will be on security and usability reviews. During the operating phase, the main function will be maintaining the data in the project.

The data integrity in the project should be maintained during the project's lifecycle to keep track of the changes and decisions made on previous stages. The configuration management is the responsible author for keeping track of the changes made on the project and freezing the configuration baselines for future review.

Configuration management

Configuration management is in an important role during a complex engineering project. Configuration management's main function is to establish consistency of the project's requirements, design, and operational information throughout its lifecycle. The configuration management approach is often used in software projects, but in building projects, there is an additional element added when physical objects are connected to the process. Configuration baselines are used to formalise specific points in the project, which can be later revisited to inspect decisions made at that time. Leningrad II nuclear power plant in Russia is used as a reference plant and design documents from that plant are partially used in baseline 0, which is the first frozen baseline.

5.2 Plant breakdown structure and data relationships

Plant Breakdown Structure (PBS) is a division of the nuclear power plant into functional systems and structures. PBS is created for organizing activities for nuclear power plant construction project and it allows the linking of the design bases with specific structures, systems and components in order to provide traceability during the project's implementation and subsequent power plant life cycles. Figure 3. below illustrates the power plant's breakdown structure. It has a plant level, an architecture level, a system building level and an equipment component material level. The Geographical Breakdown Structure (GBS) defines the buildings, elevation inside the building and rooms that belong to those elevations. SSC is an abbreviation used for Structures, Systems and Components.

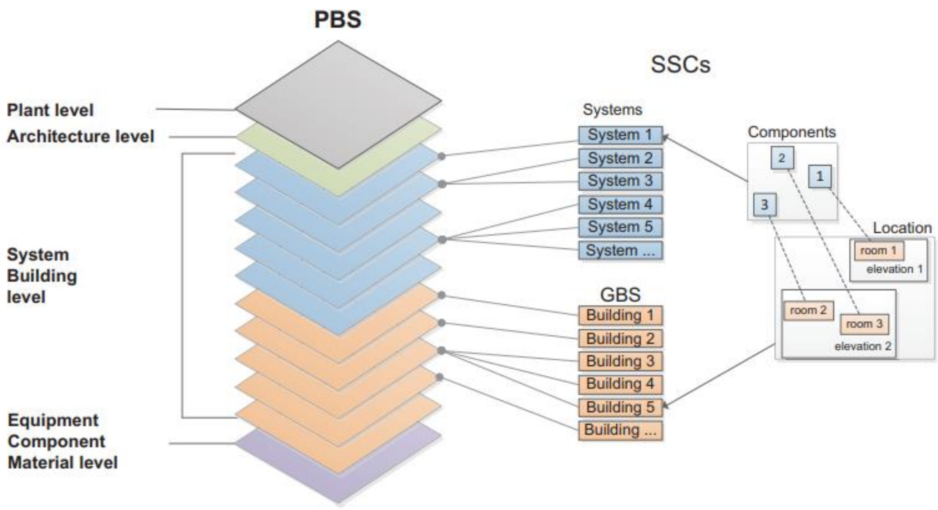


Figure 3. Plant breakdown structure.

Data model

The basic data relationship model of the power plant is illustrated below in Figure 4. The plant itself is on the top and it’s connected to a location. The location has two main branches that are connected to the systems and buildings. The building side of the model has building connected to elevation and room, which has a fire compartment. This can be also found in the plant breakdown structure. Finally, all the objects are connected to a tag which is defined for all system buildings and equipment. Every physical object in the plant design is defined by a unique identifier, including pipelines, valves, and ventilation channels. For the classification task of this study the room data that is seen in the figure is used. The rooms of a few buildings are included in that data set. The other branch of the location is connected to systems and PBS codes. Every system has a document master, which is a list defining the design documents belonging to that system. Documents have a version, which belongs to a specified configuration baseline. Preliminary Safety Analysis Report (PSAR) is presenting the design criteria and information for the nuclear reactor. The final section in the relationship description is the design package information. Design packages contain a set of similar documents that are reviewed as a package. The idea of design packages is that the design documents of the same topic can be grouped. However, the design packages were planned poorly, and they can contain documents that are not closely related. It created another issue with the review, which is presented later in this chapter.

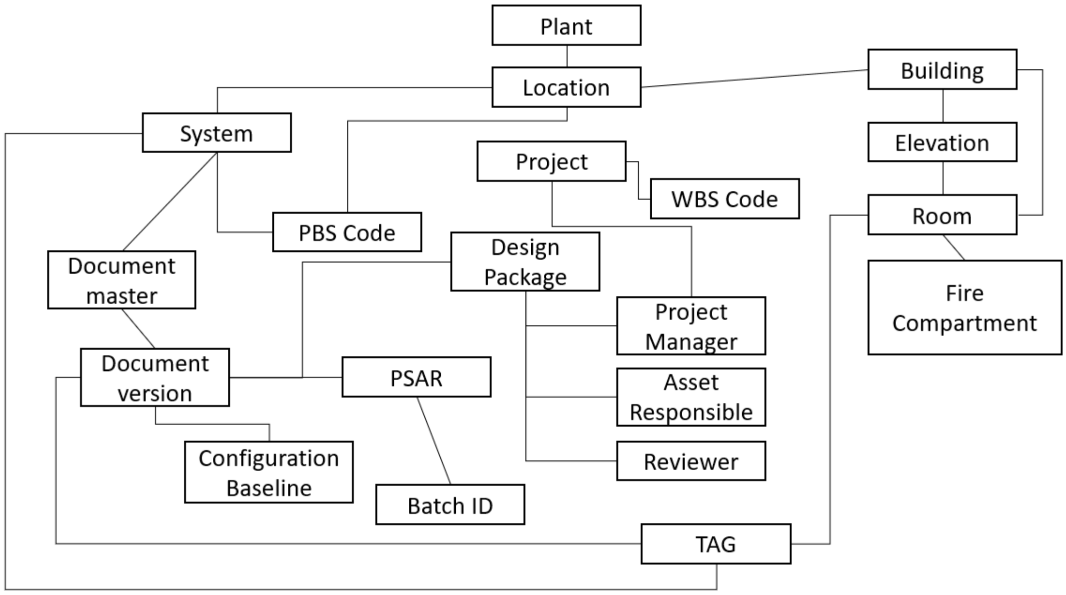


Figure 4. Data Relationships.

5.3 Data description

Design documents contain the plant design. The Transmittal flow process in Figure 5. describes how design documents and reference data are loaded into the data management system. The design documents are sent to the product owner in transmittals that are received in M-Files. These transmittals contain the design documents, DHP Files and responses to changes that were requested for previous revisions. The plant supplier delivers transmittals first to the supplier vault. In the supplier vault, a document coordinator is checking that the transmittal contents are matching. If everything in the transmittal is approved, it is sent to the FH1 vault. In the FH1 vault, the metadata of the design document is checked, and a decision is made based on that. Most of the documents are transferred to IRT and some are reviewed in M-Files.

The plan was to review all the design documents in IRT, but there were multiple reasons why it was not possible. The first reason was that the package-based review was not implemented correctly. A decision was made that the design documents belong to a package, which is reviewed in IRT. The plant supplier who was responsible for creating these design packages didn’t follow a good practice when deciding which documents belong to a specific package. This caused the packages containing documents from multiple assets and it made the review process impossible in IRT because one review package could only be assigned to one asset responsible. A solution for that problem was introduced during the basic design phase. The design packages were split into sub-packages so that one sub-package was only containing design documents that were assigned to one asset responsible. This enabled more documents to be reviewed in IRT.

The second reason was that in some disciplines, external consultants who didn't have access to IRT were reviewing the design documents. At the time, the comments for reviews could not be transferred to IRT, which made it necessary to conduct the review in M-Files for those packages. Now, this issue has been solved as transferring comments is possible. The third reason for splitting the review between two systems, was that some of the asset responsibles were not trained to perform the review in IRT. The IRT review training should have been held for those people and not allow the use of an old system. Even if the documents are reviewed in M-Files, it is still needed to load them to IRT as legacy documents because the reference data objects are related to specific document revisions. Once the detailed design phase starts, the logic behind the package grouping will change. The design documents don't have a predefined package anymore, but they will be grouped by the asset they belong to. This will fix the inconsistencies with the current logic.

Conducting the design document review concurrently in two systems raises a few issues. When different revisions of the same document are reviewed in different systems, the comments are not automatically transferred between the systems. It is a manual process to first export the comments from one system and then import them to the other systems. Since it is a manual task, there will always be some delay, which will slow down the review process. The documents that were reviewed in M-files are imported to IRT as legacy documents. DHP files of those documents were not transferred to the IRT at the same, which causes some reference data to be missing. These DHP files are imported later as a bulk import to fill the gaps in reference data.

The integration process is done by the Red Hat Fuse integration platform. It reads the document file and the DHP file and loads them to the IRT database. The Smart Data Validator (SDV) is used to validate the incoming data. The SDV is introduced later in this chapter. In IRT the documents are assigned to an Asset Responsible, who will conduct the review.

Figure 5. shows the incoming transmittal process. The first two lanes are operations that are performed in M-Files and the last lane represents the IRT environment. The transmittal process includes quite many manual steps which consumes a lot of resources. Some of these steps would be avoided completely if all the design document reviews could be done in the same system. Another issue is integration process itself. There would be no need for the integration if the persons delivering the documents could directly access the IRT environment. Now documents that are reviewed in IRT need to go through the process in M-Files, even though that has no benefit for the review itself.

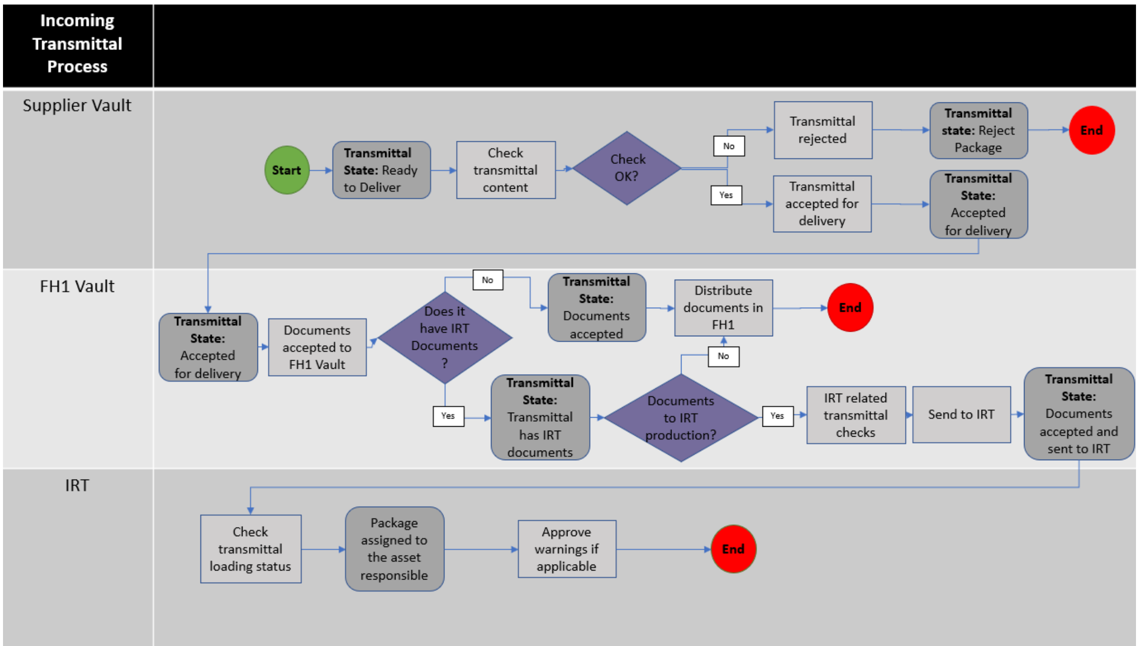


Figure 5. Transmittal flow.

Legacy data

The legacy documents are also loaded to IRT from M-Files. The legacy document in this context means a design document, that has been reviewed in M-Files before the new management system was introduced or a document that was not suitable for IRT due to reasons discussed later. These documents are needed in the IRT to create the correct relationships to data objects. The problem with legacy documents comes from outdated and missing data. The metadata for M-Files documents is not updated frequently, which causes information decay when time passes. The metadata attributes also differ from the ones in IRT. Therefore, some information is always lost when documents are moved from one system to another. The legacy document import process starts by checking which document revisions are missing from the IRT and available in M-Files. After the missing revisions are defined, those documents are exported from M-Files. The metadata sheet is then modified to match the required attributes for IRT import. The import process is done by the legacy document importer. The documents files and metadata attributes are validated with Smart Data Validator, after which they are loaded into the system.

Reference data

Reference data can be seen as base data in the management system, and it's not related to a specific area. The main function of reference data is to categorize the data and create relationships between the data objects. The reference data objects should be imported into the system before the design document transmittals are loaded. Doing the loading in this order creates the relationships between the document and reference data objects. If some data that is referenced in the document is missing, a validation report informs the user about the missing objects. In the case of design document transmittals, this will not prevent the document from loading, but it will stop the DHP data load. At the time when most of the design document transmittals were loaded into the system, there was quite a lot of missing reference data. Since the priority is to start the review process as fast as possible, a decision was made that the transmittals should be loaded into the system without creating relationships to the referenced data objects. After the reference data is up to date, the DHP files will be loaded again to the system and missing relationships are established. The sequence in reference data loading is important for establishing the relationships correctly. If a referred data object is missing from the system, the new object cannot be loaded into the system. Sometimes there is a delay in receiving some of the reference data from the plant supplier, which will then impact the import process of other data as well.

Requirements

Requirements can be seen as a part of reference data. Fortum ADLAS – Advanced Licensing and Safety Design method for Nuclear Facilities is used for handling the requirements. It is a systematic configuration management method for formulating nuclear plant requirements. ADLAS delivers a complete, unified, and hierarchical licensing documentation with traceability from plant-level requirements to equipment level specifications (Fortum, Services). The ADLAS requirements are referred to in the design documents and those requirements can be backtracked to the native requirements. The native requirements consist of YVL requirements, standards of the nuclear power industry and EPC requirements. YVL requirements come from the Finnish Regulatory Guides on Nuclear Safety and Security. These YVL requirements are assessing several topics regarding to safety management of a nuclear facility, plant and system design, radiation safety, nuclear waste, and equipment of a nuclear facility. EPC requirements are from the engineering, procurement, and construction contract.

DHP files

The DHP files contain the information that is defined in the design document that it is related to. It includes the scope, inputs, defined items, relations and attributes of the document. The scope tells the related document’s state in the configuration baseline where it belongs. The inputs are requirements, which this document fulfils. Defined items can be configuration items or other data objects that the document defines. In relations, all of the related documents are listed. The attributes tell in which area, seismic class and safety class the document belongs. The main purpose of the DHP files is to define the reference data that is included in the design document itself. The information is not needed when conducting the review of the document, but it is used to populate the data model and create relations between the objects.

5.4 Smart Data Validator

Smart Data Validator is comprehensive data validation, transformation, and migration platform, that enables the users to ensure the quality of information and reduce the time and costs associated with data take-on. Smart Data Validator (SDV) supports the import, validation, and export of data, providing consistency and traceability of the entire data process. (Intergraph, 2015).

SDV validates the data that is loaded to the plant data management system. It does not allow the user to load data that is invalid. First, all of the data is loaded into a staging area. At this point, only the number of columns in the csv file and the names of those columns are checked. All data values are accepted in this step. When the data is in the staging area, the validation rules are run. After the validation has been done, there will be a validation report that states any conflicts with the rules. If there are no validation errors or warnings, the exporting process will start automatically. If there are conflicts with the validation rules, a validation report will show these as errors or warnings. Errors should always be corrected before continuing the export process. When faulty data is loaded into a system, the correct relationships between data objects are not created and the data objects are not usable for upcoming processes. The warnings shown in the validation report are usually referring to missing documents, and they can be ignored since the referred documents are sometimes not available at the time. When the validation errors and warnings are either fixed or approved, the extraction process can start, and data will be loaded into the system.

There are multiple types of validation rules in the system, which are used in different scenarios. These rule types are presented below:

1. Objects exist check
2. Format check
3. Measurement check
4. Cardinality check
5. Picklist values

The Objects exist check is checking if the data objects that are referenced by the new object are found in the system. For example, if a tag that belongs to a document is tried to be loaded into a system, the SDV will perform a check if the document exists. If it doesn’t, the loading process is paused and a validation report is created, which will tell the user what document is missing from the system. These can sometimes be ignored if it’s known that the missing data will be loaded later.

The format check is used to ensure that the value is formatted correctly. For example, a date format can be defined as “dd.mm.yyyy” and if the date is not in that format, the loading process is paused and the value needs to be corrected. Formatting errors usually occur in date values and Preliminary Safety Analysis Report codes. The incorrect code values are caused because of the different formatting styles between two different systems.

A measurement check will validate if the values are in the accepted range. It will prevent data entries that have extreme values entered by accident. This check is defined for numeric attributes only.

In a relational database, there can be three types of relationships. Those are one-to-one, one-to-many and many-to-many. The cardinality check is ensuring that these relationship rules are not violated. For example, a room can only be in one building, but one building can have many rooms. If there are invalid cardinalities, those are included in the validation report and should be fixed before the export can start.

The picklist values are set for some attributes. For example, the possible values for document delivery purposes are “For Approval”, “For Comments and “For Information. If the value is something else than one of these pre-defined values, the SDV will pause the loading process and include the issue in the validation report.

Some critical attributes are set as mandatory attributes, which means that there needs to be a value for validation to be completed. Other attributes are accepting missing values, as those are not defined as critical for the data structure. The missing values can occur because all reference data attributes are not defined for all objects. A Properly defined set of validation rules is the most important factor for a successful validation process.

The Hexagon Smart plant foundation is a powerful tool that is used in various fields for data management. In this nuclear plant project, it was chosen to be used as a review tool, which it was not originally designed. The product was customized to support the review process, but looking back, perhaps a better solution would have been to use it only as a data management system and have a different system for document reviews. Also, the integration between M-Files and IRT is costing a lot of resources and time. If the plant supplier had access directly to the IRT, the data and design documents could be delivered straight to the system, reducing the possibility of data corruption.

6 Classification for reference data

This chapter focuses on the reference data that is used in the data management system. The reference data will be analysed to find out if there is any missing data. After that, a classification algorithm is applied to the data set to find out if it's possible to build a model to predict missing attribute values for data instances. The goal is to find a classification model that can be used for data imputation.

6.1 Classification data

The data used for building the classification model consists of 32 attributes, including nominal, ordinal, continuous, binary and discrete values. The attributes belong to a room, which is a part of a building in the plant breakdown structure. The attributes describe the characteristics of the room, including physical properties like building materials and fire extinguishing agents, and nonphysical properties, like safety divisions and radiation zones. Responsible discipline is chosen as a class attribute for the model. It has seven possible values, which is a suitable amount for the predictive model. These seven values are Architecture, Electrical, HVAC, Instrumentation, Mechanization and Fuel handling, Process and Reserve. They tell in which responsible area the room belongs in the system. The data set is exported from the Nawisworks design tool, which is used for 3D modelling the power plant. Other reference data is sent by the plant supplier and is then imported to the IRT, but room data must be exported from the design tool. The Navisworks room data was chosen because it has more attributes and is more complete than the room data in IRT. The data set has 546 instances, and it consists of rooms from three different buildings.

6.2 Classification algorithm selection

The classification model is built with Waikato Environment for Knowledge Analysis (Weka) software. Weka has a large variety of machine learning and data mining algorithms, and tools for visualizing and analysing data. The tool is free to use and is based on Java programming language. It also has the ability to pre-process the data.

The algorithm used for building the classification model is J48, which is also known as C4.5. It is a classification algorithm, which constructs a decision tree The J48 implementation of the C4.5 algorithm has additional features that are helpful when operating with real-world data. The algorithm can operate with missing values, and it splits continuous attributes into value ranges. The data used for building the model has nominal and numeric attributes, which makes J48 a good choice for this dataset.

Before performing any data pre-processing, a classification model is created, to compare it with the final model. The goal of pre-processing is to increase the model's accuracy, make it more efficient for the computing system and make it easier to interpret for data imputation task. Table 4. shows the accuracy of the model with all attributes.

Table 4. J48 with all attributes.

	J48
Classification Accuracy	86.3 %
Number of leaves	56
Size of the tree	84
Confidence factor	0.25
Minimum number of objects per leaf	2

The classification accuracy is 86.3 % in the first iteration. The decision tree produced by the J48 algorithm is quite complex when all the features are present. The confidence factor and minimum objects per leaf are at their default values. The confidence factor is used for pruning the decision tree. A smaller confidence factor value incurs more pruning, which reduces the chance of overfitting the model. Different confidence factor values will be tested for optimal performance at a later stage. The minimum number of objects represents the minimum size of the nodes. A greater number indicates more pruning since the node splitting is stopped earlier. Increasing this value also reduces the chance of overfitting the model and different values will be experimented later.

6.3 Pre-processing

The first step of pre-processing is reducing the number of attributes to simplify the model. Feature selection is first done by analysing the distribution of attribute values. Name, Description and KKS code of the compartment are deleted because they contain unique values or too many distinct values, which makes them unusable for model building. Availability of automatic fire detection system, Availability of alarm and evacuation control system, Availability of sensor for continuous monitoring of doors and hatches position in safety division and Explosion hazard zone are removed, because their attribute values are too skewed towards one value. The remaining data is then imported to Weka for the next step of pre-processing. These removed attributes are presented in Table 5.

Table 5. Removed attributes.

Attribute Name	**Attribute Type**	**Reason for excluding**
Name	Nominal	Name has unique values, not useful for the model
Description	Nominal	Too many different values
KKS code of fire compartment	Nominal	Too many different values
Availability of automatic fire detection system	Binary	Almost all instances have the same value
Availability of alarm and evacuation control system	Binary	Almost all instances have the same value
Availability of sensor for continuous monitoring of doors and hatches position in safety division	Binary	Almost all instances have the same value
Explosion hazard zone	Nominal	Almost all instances have the same value

After the seven attributes presented above are deleted, there are 25 attributes remaining. These attributes and their data type are listed in Table 6.

Table 6. Attributes imported to Weka.

Attribute Name	**Attribute Type**
Access Area	Nominal
Radiation Zone	Ordinal
Radiation zone at PPM	Ordinal
Full fire load in the room	Continuous
Fire load per unit area for the room	Continuous
Fire load category	Ordinal
Responsible discipline	Nominal, class attribute
Safety division	Nominal
Fire hazard class	Nominal
Availability of automatic fire extinguishing system	Binary
Type of fire extinguishing agent in automatic fire extinguishing system.	Nominal
Annunciator type of alarm and evacuation control system	Nominal
Type of AFDS detector	Nominal
Availability of smoke protection system	Binary
Number of fire extinguishers	Discrete
Weight of extinguishing agent	Continuous
Noise level zone	Ordinal
Noise level	Discrete
Volume	Continuous
Floor Area	Continuous
Ceiling area	Continuous
Wall area	Continuous
Interior floor finishing	Nominal
Interior ceiling finishing	Nominal
Interior wall finishing	Nominal

When the data has been imported to Weka, the next step is to find if there are any extreme values or outliers. For categorical attributes, the possible outliers are mostly contextual outliers and finding them would require in-depth knowledge about the attributes and what are the relationships between them. Analysing those relationships is outside of the scope of this thesis and these possibly occurring contextual outliers, or anomalies will not be touched in this classification process. When it comes to missing values in the data set, there are some attributes that have blank values, but those are almost all features that simply do not exist in the corresponding rooms. These blank instances should not be removed from the data set but treated as a valid attribute value.

At this point, there are still 25 attributes left, which is too much for a data set of this size. The benefit of feature reduction is that the model becomes simpler to understand and more efficient to execute when running the algorithm multiple times with different parameters. The feature reduction is done by testing a few different ranking methods and selecting the removed attributes based on those. The correlation coefficient and Information gain are calculated with Weka.

Figure 6. shows the Pearson's correlation results. The correlation is calculated between each attribute and the class. The Attributes are ranked by their correlation with the class, values close to 1 have high correlation and values close to 0 have low correlation. A threshold needs to be set to filter attributes that are below the threshold.

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 Responsible_discipline):
        Correlation Ranking Filter
Ranked attributes:
 0.3875    9 Availability_of_automatic_fire_extinguishing_system
 0.3864   10 Type_of_fire_extinguishing_agent_in_automatic_fire_extinguishing_system
 0.3353    4 Fire_load_category
 0.3101   20 Noise_level
 0.301    21 Noise_level_zone
 0.2983   13 Type_of_AFDS_detector
 0.2664    1 Access_area
 0.2562   27 Interior_ceiling_finishing
 0.2562   28 Interior_wall_finishing
 0.2506    8 Fire_load_per_unit_area_for_the_room
 0.2361   18 Type_of_extinguishing_agent
 0.2281   26 Interior_floor_finishing
 0.2173   14 Availability_of_smoke_protection_system
 0.208    12 Annunciator_type_of_alarm_and_evacuation_control_system
 0.1753    3 Radiation_zone_at_PPM
 0.1743    2 Radiation_zone
 0.1537   16 Availability_of_fire_extinguishers
 0.1535   15 Type_of_smoke_protection_system
 0.1501   17 Number_of_fire_extinguishers
 0.104    22 Floor_area
 0.104    23 Ceiling_area
 0.0978   24 Wall_area
 0.079     6 Safety_division
 0.0751   11 Full_fire_load_in_the_room
 0.0664    7 Fire_hazard_class
 0.0624   25 Volume
 0.051    19 Weight_of_extinguishing_agent
```

Figure 6. Attributes ranked by Pearson's correlation.

The correlation values are somewhat equally distributed and there is no clear point where the information significantly decreases. For this situation 0.25 is decided to be a reasonable cut-off point.

The Information gain is the second feature selection algorithm selected for the data. In Figure 7. the attributes are ranked by their Information gain. The information gain threshold for feature selection is set to 0.5, which leaves 12 attributes above the threshold.

Attribute selection output

```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 Responsible_discipline):
        Information Gain Ranking Filter

Ranked attributes:
 1.293   21 Noise_level_zone
 1.216   20 Noise_level
 0.969    8 Fire_load_per_unit_area_for_the_room
 0.792   19 Weight_of_extinguishing_agent
 0.594   10 Type_of_fire_extinguishing_agent_in_automatic_fire_extinguishing_system
 0.594    9 Availability_of_automatic_fire_extinguishing_system
 0.594   27 Interior_ceiling_finishing
 0.594   28 Interior_wall_finishing
 0.594   13 Type_of_AFDS_detector
 0.587    4 Fire_load_category
 0.551   23 Ceiling_area
 0.551   22 Floor_area
 0.43    11 Full_fire_load_in_the_room
 0.42    26 Interior_floor_finishing
 0.368    3 Radiation_zone_at_PPM
 0.36     2 Radiation_zone
 0.29    18 Type_of_extinguishing_agent
 0.283   17 Number_of_fire_extinguishers
 0.262   25 Volume
 0.243   12 Annunciator_type_of_alarm_and_evacuation_control_system
 0.236   24 Wall_area
 0.219    1 Access_area
 0.191   16 Availability_of_fire_extinguishers
 0.18    14 Availability_of_smoke_protection_system
 0.118   15 Type_of_smoke_protection_system
 0.109    6 Safety_division
 0        7 Fire_hazard_class
```

Figure 7. Attributes ranked by Information Gain.

Table 7. Most Informative attributes achieved by both methods.

Rank	**Pearson's correlation**	**Information Gain**
1	Availability of automatic fire extinguishing system	Noise level zone
2	Type of fire extinguishing agent in automatic fire extinguishing system	Noise level
3	Fire load category	Fire load per unit area for the room
4	Noise level	Weight of extinguishing agent
5	Noise level zone	Type of fire extinguishing agent in automatic fire extinguishing system
6	Type of AFDS detector	Availability of automatic fire extinguishing system
7	Access area	Interior ceiling finishing
8	Interior ceiling finishing	Interior wall finishing
9	Interior wall finishing	Type of AFDS detector
10	Fire load per unit area for the room	Fire load category
11	-	Ceiling area
12	-	Floor area

Both ranking methods achieve virtually the same results. Table 7. shows selected attributes from both ranking methods. Interior ceiling finishing and Interior wall finishing always have the same values in the data set, so only one of them should be used. The same situation occurs with Ceiling area and Floor area. The final set of attributes for the model building contains Availability of automatic fire extinguishing system, Type of fire extinguishing agent in automatic fire extinguishing system, Fire load category, Noise level, Noise level zone, Type of AFDS detector, Interior ceiling finishing and Fire load per unit are for the room.

After removing the uninformative features from the data set, the J48 classification is carried out again. The algorithm parameters have the same values as in the first iteration. Table 8. shows the results from first and second iterations.

Table 8. Model accuracy after reducing features.

	J48	**J48**, selected features
Classification Accuracy	86.3 %	85.3 %
Number of leaves	56	52
Size of the tree	84	79
Confidence factor	0.25	0.25
Minimum number of objects per leaf	2	2

After reducing the less informative attributes, the classification accuracy decreases from 86.3 % to 85.3 %. The size of the decision tree and the number of leaves in the tree decreased slightly. The next improvement for the model is discretizing the continuous attributes. Even though the J48 algorithm can deal with continuous values, discretizing these attributes beforehand helps to reduce the size of the tree, making the model simpler. Discretising the continuous attributes manually will improve the algorithm efficiency since the splitting points can be optimized. The discretization can be done by dividing the continuous attributes into equal sizes of bins or bins with equal ranges. The equal size of bins is a better solution in this case since an equal range would lead to unbalanced discretisation due to the distribution of values.

The effects of changing the confidence factor to a lower value and increasing the minimum node size are presented in Table 9. below. When the decision tree complexity decreases, the model accuracy also decreases.

Table 9. J48 parameters.

	J48, selected features	**J48**, Discretised attributes	**J48**, Discretised attributes	**J48**, Discretised attributes
Classification Accuracy	85.3 %	85.2 %	82.0 %	82.6 %
Number of leaves	67	49	30	39
Size of the tree	101	69	38	54
Confidence factor	0.25	0.25	0.25	0.1
Minimum number of objects per leaf	2	2	10	2

A reasonable choice is to pick the model with discretised attribute values. It has an accuracy of 85.2 % and the decision tree is not too complicated. Decision trees with more pruning achieve a simpler structure, but the accuracy also starts to decrease. The most common class attribute value in the data set is occurring in 34 % of the instances. Using the classification model result is much more precise than choosing the most common value for Responsible discipline. It means that there is a connection between the attributes, and they are not just independent variables. Based on this information the classification task could be extended to cover other attributes as well. This classification model was built for one attribute for the building data set. There is a lot more reference data in the system, where these techniques could be applied to.

The confusion matrix is presented in Table 10. It tells how many of each class attribute values are assigned correctly and how many aren't. It also shows which classes are incorrectly assigned to other classes. The diagonal values that are highlighted, are the correctly classified instances. For example, the Process discipline is correctly classified 141 times and incorrectly classified 40 times. The small disciplines, like Mechanization and Reserve, have only incorrectly classified instances. If a room is classified to the Electrical discipline, there is a 97.2 % chance that the classification is correct. The poor classification accuracy in some classes should be taken into consideration when performing imputation for the data set.

Table 10. Confusion matrix.

Classified as →	**A**	**B**	**C**	**D**	**E**	**F**	**G**
A = process	**141**	0	16	8	1	12	3
B = Mechanization	2	**0**	2	0	1	0	0
C = Architecture	2	0	**108**	0	1	3	0
D = Instrumentation	0	0	0	**26**	0	0	0
E = Electrical	0	0	1	0	**104**	2	0
F = HVAC	8	0	1	0	0	**86**	1
G = Reserve	10	0	0	1	0	6	**0**

The decision tree of the final model is presented in Appendix 1. The noise level zone is the root node. The tree structure is divided into three figures since the tree is quite large. That is caused by the amount of class attribute values, which was quite high for a classification task. The square nodes show the ratio of correctly classified instances in each case.

7 Conclusions

The objective of this study was to identify the data integrity issues in the project and in the data management system. First, the study gave an overview of data quality, which is a key part of data integrity. The second part was introducing the machine learning method called classification, which was used to create a classifier for missing data imputation. The third part explained the case company’s data management system and identified the problems with the current system. After that, a classification algorithm was used to create a model for data imputation for a data set. The answers this study provides for research questions are as follows:

1. *How are the problems in data integrity affecting the review process?*

The Smart Plant Foundation (SPF) is a powerful tool as a plant information management system, but it is not designed to be a specific review tool. Due to the limitations of the tool and the poor design of the review packages, the review could not be conducted in IRT for all documents. For this reason, some of the documents are being reviewed in M-Files. When the same document has different revisions reviewed in two systems, the comments need to be manually transferred to the IRT. The transmittals which are used to import the design documents into the IRT involve a lot of manual work in the M-Files. It causes delays to the review process, which could be avoided if the documents could directly be sent to the review tool.

2. *What are the causes and impacts of poor quality of data?*

The main problem with the data quality is that multiple systems are used simultaneously to handle the data and these systems are not synced. The sequence of the data loading is many times the reason for discrepancies. If a set of design documents that have a relationship to a building is received, but the document defining that building is not yet received, the relationships are not created. The reference data is currently received from multiple sources. The plant supplier is sending some of the reference data, but some needs to be exported from the 3D design system. Since there are multiple sources for the reference data, the information is not received at the same time from all sources. Due to this delay, some relationships may have already changed and some data objects may not exist anymore.

The impacts of poor data quality are diverse. There is a monetary cost of people working to fix the data issues, which is hard to estimate. If the review task is assigned to the wrong person, it will delay the review. Communicating with the plant supplier and finding the cause of data issues is time-consuming. Incorrect decisions can be made if the data contains inaccurate information.

3. *How machine learning can be used to improve data?*

In this study, a supervised machine learning method, classification, was used for a data imputation task. Two different classification algorithms were introduced and J48 was chosen for the model building. The data set was a room reference data set, containing categorical and numerical attributes. The goal was to find out if it's possible to build an accurate model for that data and could the missing data values be predicted from the existing data instances. The results show there is a connection between the attributes and missing values can be predicted from other values.

This study focused on one data set, which does not give a full understanding of all the available data on the project. The next step is to continue experimenting with the classification methods on other reference data found in the system. There are several classification algorithms that can be tested to find the best fitting solution for each data set. Other machine learning techniques could be also implemented to study the available data. For example, anomaly detection is an interesting topic that can be used to find abnormalities from the data.

References

Affendey, L., Ibrahim, H., Jabar, M., Mustapha, A., Panahy, P., Sidi, F. 2012. Data quality: A survey of data quality dimensions. International Conference on Information Retrieval & Knowledge Management. pp, 300-304.

Ali, S., Ao, Y., Li, H., Zhongguo, Y. 2017. Choosing Classification Algorithms and Its Optimum Parameters based on Data Set Characteristics. Journal of Computers. Vol. 28, No. 5, pp. 26-38.

Bah, M., Hammad, M., Wang, H. 2019. Progress in Outlier Detection Techniques: A Survey. IEEE Access. Vol. 7, pp. 107964-108000.

Batini, C. & Scannapieco, M. 2016. Data and Information Quality: Dimensions, Principles and Techniques. Springer International Publishing, Cham.

Bernardino, J., Laranjeiro, N., Soydemir, S. 2015. A Survey on Data Quality: Classifying Poor Data. The 21st IEEE Pacific Rim International Symposium on Dependable.

Bernardino, J., Laranjeiro, N., Soydemir, S. 2015. A Survey on Data Quality: Classifying Poor Data. The 21st IEEE Pacific Rim International Symposium on Dependable.

Blake, R. & Mangiameli, P. 2008. Proceedings of the 13th International Conference on Information Quality, MIT, Cambridge.

Bogunovic, N., Brkic, K., Jovic. 2015. A review of feature selection methods with applications. 38th International Convention on Information and Communication Technology, Electronics and Microelectronics. Pp. 1200-1205.

Boone, C., Hazen, B., Ezell, J., Jones-Farmer, L. 2014. Data quality for data science, predictive analytics, and big data in supply chain management: An introduction to the problem and suggestions for research and applications. International Journal of Production Economics. Vol. 154, pp. 72-80.

Bosch, K., Fursova, N., Gelsema, T., Giessing, S., Guarnera, U., Loo, M., Petrauskiene, J., Quensel von Kalben, L., Scanu, M., Walsdorfer, K. 2016. Methodology for data validation 1.0.

Brownlee, J. How to Choose a Feature Selection Method For Machine Learning. 2019. [online article]. Available at: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/ (accessed 15 May 2022).

Cheng, D., Li, X., Zhang, S., Zhu, X., Zong. Learning k for kNN Classification. ACM Transactions on Intelligent Systems and Technology. Vol. 8, No. 3, pp. 1-19.

Economic Commission for Europe of the United Nations (UNECE). 2000. Glossary of Terms on Statistical Data Editing. Conference of European Statisticians Methodological material, Geneva.

Elmagarmid, A., Ipeirotis, P., Verykios, V. 2007. Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering. Vol. 19, No. 1.

Estevez, P. & Vergara, J. 2014. A Review of Feature Selection Methods Based on Mutual Information. Neural Computing and Applications. Vol. 24, pp. 175-186.

Fortum. ADLAS® - Advanced Nuclear Licensing and Safety Design Method for Nuclear Facilities. [online article]. Available at: https://www.fortum.com/products-and-services/power-plant-services/nuclear-services/nuclear-newbuild-and-upgrades-services/adlasr-advanced-nuclear-licensing-and-safety-design-method-nuclear-facilities. (accessed 17 June 20222).

Gao, Y. & Wang, H. 2021. Research on C4.5 algorithm improvement strategy based on MapReduce. Procedia Computer Science. Vol. 183, pp. 160-165.

Gao, Z., Hu, C., Wu, J. 2009. An Empirical Study on Several Classification Algorithms and Their Improvements. Lecture Notes in Computer Science. Vol. 5821.

Ghost, J., Hand, D., Kumar, V., Liu, B., Mclachlan, G., Motoda, H., Ng, S., Quinlan R., Steinbach, M., Steinberg, D., Wu, X., Yu, P., Zhou, Z. 2007. Top 10 algorithms in data mining. Knowledge and Information Systems. Vol. 14, No. 1.

Hardt, M., Fridovich-Keil, S., Recht, B., Roelofs, R., Schmidt, L., Shankar, V. 2019. A Meta-Analysis of Overfitting in Machine Learning. 33rd Conference on Neural Information Processing Systems, Vancouver.

Hu, L., Huang, M., Ke, S., Tsai, C., 2016. The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus. Vol. 5, No. 1304.

Jordan, M. & Mitchell, T. 2015. Machine learning: Trends, perspectives, and prospects. Science. Vol. 349, pp. 255-260.

Juhola, M. & Laurikkala, J. 2011. Missing values: how many can they be to preserve classification reliability? Artificial Intelligence Review. Vol. 40, No. 3, pp. 231–245.

Kotsiantis, S., Pintelas, P., Zaharakis, I. 2006. Machine learning: a review of classification and combining techniques. Artificial Intelligence Review. Vol. 26, No. 3, pp. 159-190.

Lagoze, C. 2014. Big Data, data integrity, and the fracturing of the control zone. Big Data & Society. Vol. 1, No. 1, pp. 1-11.

Lee, Y., Pipino, L., Wang, R. 2002. Data Quality Assessment. Communications of the ACM. Vol. 45, No. 4, pp. 211-218.

Lewis, P., Saunders, M., Thornhill, A. 2015. Research Methods for Business Students, 7th Edition.

Lindkvist, C., Stasis, A., Whyte, J. 2013. Configuration Management in Complex Engineering Projects. Procedia CIRP. Vol. 11, pp. 173-176.

Liu, H., Refaeilzadeh, P., Tang, L. 2016. Cross-Validation. Encyclopedia of Database Systems. Springer, New York.

Mangal, A., & Holm, A. (2018). A Comparative Study of Feature Selection Methods for Stress Hotspot Classification in Materials. Integrating Materials and Manufacturing Innovation. Vol. 7, pp. 87-95.

Olson, J. 2003. Data Quality: The Accuracy Dimension. Elsevier Science & Technology, San Francisco.

Quinlan, J. 1993. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, Inc. Mach Learn 16, pp. 235–240.

Ruggieri, S. 2002. Efficient C4.5 [classification algorithm]. IEEE Transactions on Knowledge and Data Engineering. Vol. 14, No. 2, pp. 438-444.

Sebastian-Coleman, L. 2013. Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework. Elsevier Science & Technology, San Francisco.

Sharma, A. 2021. How to Ensure Data Integrity in Your Organization [online article]. Available at: https://www.dataversity.net/how-to-ensure-data-integrity-in-your-organization/. (accessed 26 May 2022).

Simon, A. 2013. Definition of validation levels and other related concepts**.** European Commission Eurostat/B1, Eurostat/E1, Eurostat/E6.

Simon, R. 2007. Resampling Strategies for Model Assessment and Selection. Fundamentals of Data Mining in Genomics and Proteomics. Pp. 173-186.

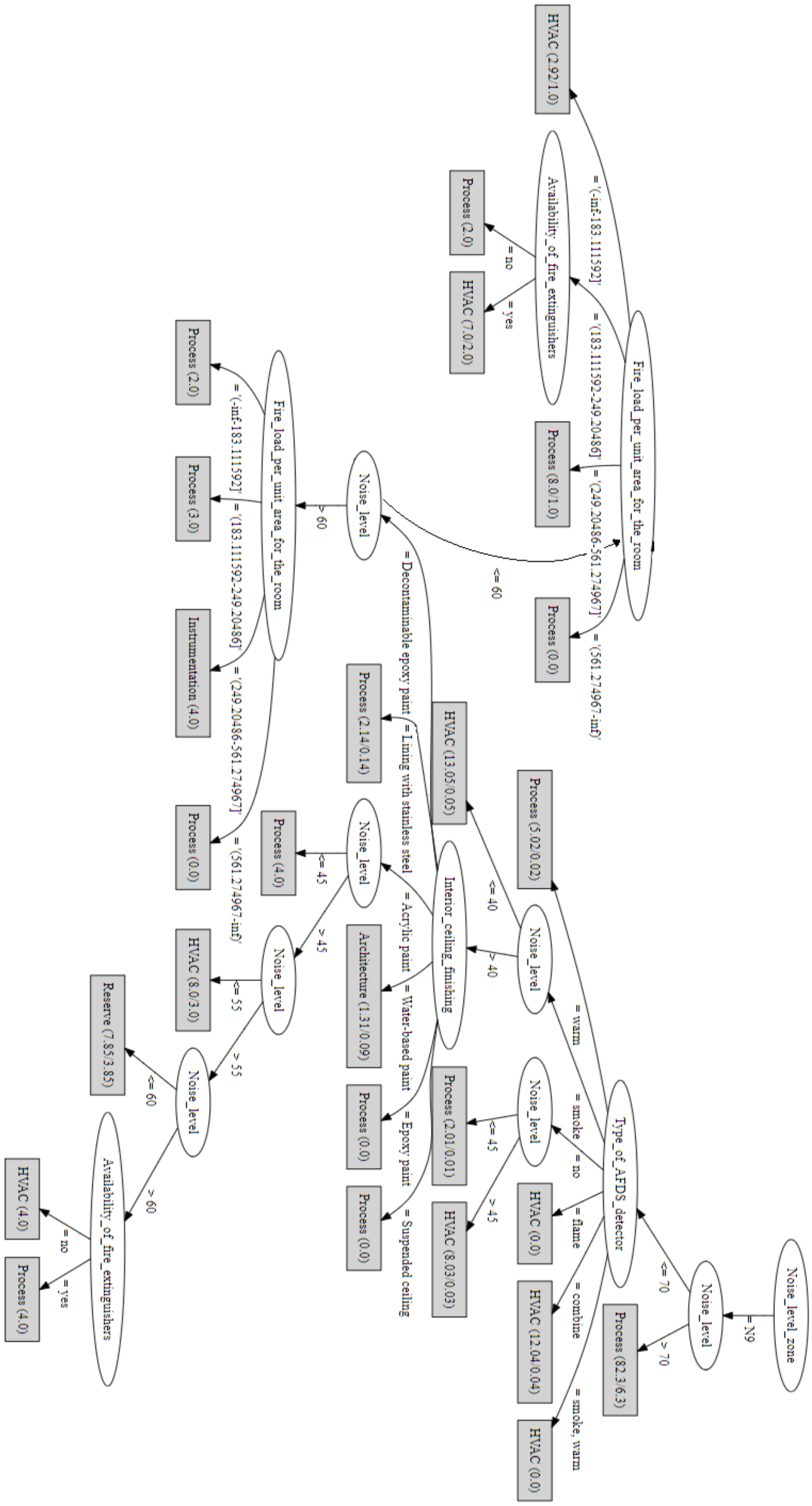
Wang, R. & Strong, D. 1996. Beyond Accuracy: What Data Quality Means to Data Consumers. Journal of management information systems. Vol. 12, No. 4, pp. 5–33.

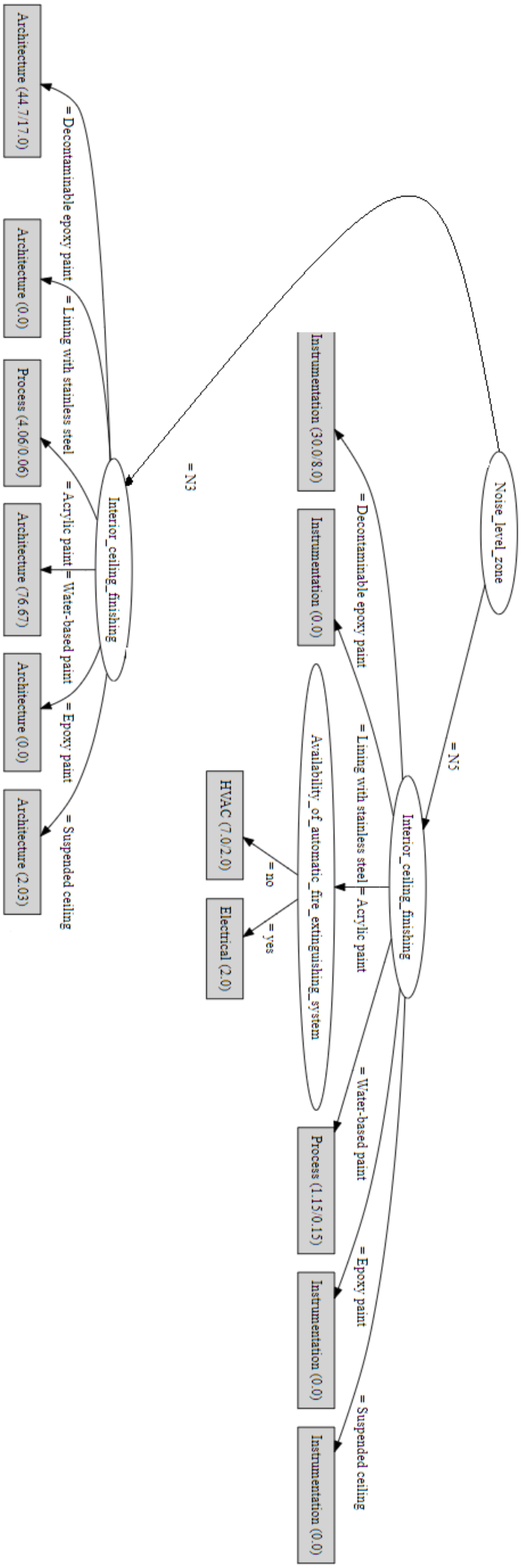
Weka Sourceforge. Available at: https://weka.sourceforge.io/doc.stable/. (accessed 16 May 2022).

Yin, R. 1981. The Case Study as a Serious Research Strategy. Knowledge: Creation, Diffusion, Utilization. Vol. 3, No.1, pp. 97-114.

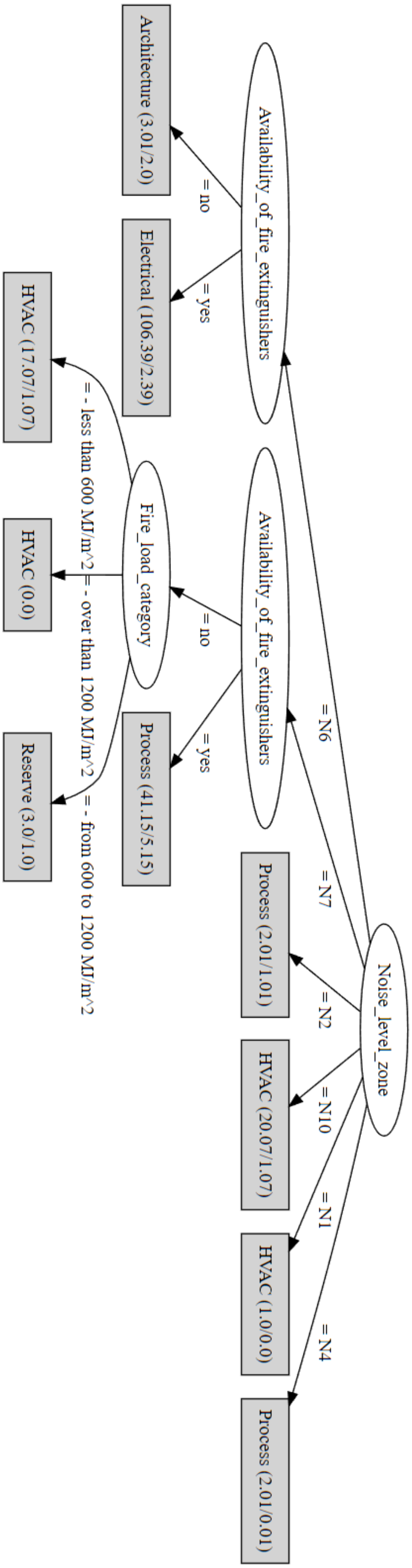
Ying, X. 2019. An Overview of Overfitting and its Solutions. Journal of Physics: Conference Series. Vol. 1168, No. 2.

Appendix 1. Decision tree of the classification model





Noise_level_zone
= N3
= N5
Interior_ceiling_finishing
= Decontaminable epoxy paint
= Lining with stainless steel
= Acrylic paint
= Water-based paint
= Epoxy paint
= Suspended ceiling
Architecture (44.7/17.0)
Architecture (0.0)
Process (4.06/0.06)
Architecture (76.67)
Architecture (0.0)
Architecture (2.03)
Interior_ceiling_finishing
= Decontaminable epoxy paint
= Lining with stainless steel
= Acrylic paint
= Water-based paint
= Epoxy paint
= Suspended ceiling
Instrumentation (30.0/8.0)
Instrumentation (0.0)
Availability_of_automatic_fire_extinguishing_system
= no
= yes
HVAC (7.0/2.0)
Electrical (2.0)
Process (1.15/0.15)
Instrumentation (0.0)
Instrumentation (0.0)


Noise_level_zone
= N6
= N7
= N2
= N10
= N1
= N4
Availability_of_fire_extinguishers
= no
= yes
Architecture (3.01/2.0)
Electrical (106.39/2.39)
Availability_of_fire_extinguishers
= no
= yes
Fire_load_category
Process (41.15/5.15)
= - less than 600 MJ/m^2
= - over than 1200 MJ/m^2
= - from 600 to 1200 MJ/m^2
HVAC (17.07/1.07)
HVAC (0.0)
Reserve (3.0/1.0)
Process (2.01/1.01)
HVAC (20.07/1.07)
HVAC (1.0/0.0)
Process (2.01/0.01)