UNIVERSITY OF OULU
FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Riikka Valkama

BEAMFORMING ANALYSIS USING RANDOM FOREST CLASSIFIER

Master's Thesis
Degree Programme in Computer Science and Engineering
May 2021

Valkama R. (2021) Beamforming Analysis Using Random Forest Classifier. University of Oulu, Degree Programme in Computer Science and Engineering, 53 p.

ABSTRACT

Wireless communication has a long history that has changed shape throughout the centuries, from smoke signals to electromagnetic radiation. Data transmission evolution has made worldwide communication possible and has contributed to globalisation. Today, information can be shared in real-time—for example, to the other side of the world. Wireless communication has evolved to the point that real-time plays a vital role, and data loss should not occur.

For efficient wireless data transmission, a beamforming technique has been developed. This is a signal processing technique used in antennas for directional signal transmission or reception. Beamforming includes numerous variations, making the analysis of beamforming challenging. Due to its complex nature, beamforming is attempted to be understood more simply at a higher level, and for that reason, elements are listed that enable the analysis to check whether beamforming succeeded on the radio.

Machine learning is a new trend in different aspects of technology. Problems are aimed to be solved and predicted more efficiently by using suitable machine learning methods. Machine learning enables more precise analysis and error tracking, which are utilised in combination to minimise errors. Furthermore, machine learning has been integrated into various automation systems. This thesis concentrates on analysing the success of beamforming at a high level and aims to automate testing and provide feedback to radio architects who utilise beamforming. For a high-level analysis, a few criteria define the success of beamforming on the radio.

In this thesis, a machine learning pipeline is presented from prepossessing to the final model, and we demonstrate the promising results we have been able to achieve using the random forest classifier. Such promising results make it possible to continue with the beamforming classification and serve as motivation to improve and gather detailed feedback for the end-user.

Keywords: beamforming, radio, 5G, machine learning, classification, analysis

Valkama R. (2021) Keilanmuodostuksen analyysi käyttäen satunnaismetsäluokitinta. Oulun yliopisto, Tietotekniikan tutkinto-ohjelma, 53 s.

TIIVISTELMÄ

Langattomalla tiedonsiirrolla on pitkä historia, joka on muuttanut muotoaan vuosisatojen aikana savumerkeistä sähkömagneettiseen säteilyyn. Tiedonsiirron kehitys on mahdollistanut maailmanlaajuisen tiedonjaon ja edistänyt globalisaatiota. Nykyään informaatiota voidaan jakaa toisille reaaliajassa, esimerkiksi toiselle puolelle maailmaa. Langaton tiedonsiirto on kehittynyt siihen suuntaa, jossa reaaliaikaisuudella on merkittävä rooli ja datahäviötä ei saisi tapahtua.

Tehokkaaseen langattomaan tiedonsiirtoon on kehitetty keilanmuodostustekniikka. Keilanmuodostus on signaalinkäsittelytekniikka, jota käytetään antenneissa suunnatun signaalin lähettämiseen tai vastaanottamiseen. Keilanmuodostus sisältää erilaisia variaatioita, jonka vuoksi sen analysointi voi olla haastavaa. Kompleksisuuden vuoksi keilanmuodostamista pyritään ymmärtämään yksinkertaisemmin korkeammalla tasolla ja sen mukaan listaamaan asioita, joiden avulla sen onnistumista radiolla voidaan analysoida.

Koneoppimisen soveltaminen eri teknologian osa-alueilla on uusi muoti-ilmiö. Koneoppimisen avulla pyritään ratkaisemaan ongelmia tehokkaammin ja ennustamaan tapahtumia kerätyn datan pohjalta. Koneoppiminen mahdollistaa tehokkaan analysoinnin ja virheiden etsimisen niiden minimoimiseksi. Koneoppimista on alettu yhdistämään erilaisiin automaatiojärjestelmiin ja tämä työ keskittyy korkeammalla tasolla analysoimaan keilanmuodostamisen onnistumista ja pyrkii automatisoimaan testausta sekä antamaan palautetta keilanmuodostustekniikkaa hyödyntäville radion kehittäjille. Korkeamman tason analysointiin on rajattu muutamia kriteereitä, jotka määrittelevät onko keilanmuodostus onnistunut radiolla.

Työssä esitellään koneoppimisen työvaiheet datan esikäsittelystä lopulliseen malliin sekä näytämme, kuinka lupaavia tuloksia saavutimme satunnaismetsä -luokittimella. Lupaavat lopputulokset mahdollistavat työn jatkamista sekä motivoivat parantamaan ja keräämään tarkempaa palautetta loppukäyttäjälle.

Avainsanat: keilanmuodostus, radio, 5G, koneoppiminen, luokittelu, analysointi

TABLE OF CONTENTS

FOREWORD

The output of the beamforming analysis is part of the automated test environment at Nokia Solutions and Networks Oy in Oulu. I feel grateful to have had the opportunity to be a part of the telecommunication community of the company.

I wish to thank the automated test environment team for the opportunity and for gathering data for this thesis. In addition, I would like to thank my supervisor of this thesis, M.Sc Juha Hannula, for his support and productive brainstorming. Moreover, I want to thank D.Sc. Jaakko Suutala and D.Sc. Satu Tamminen for supervising and reviewing.

My greatest thanks belong to my family and friends, who motivated me to study diligently and achieve my goals.

Oulu, May 5th, 2021

Riikka Valkama

LIST OF ABBREVIATIONS AND SYMBOLS

AI	artificial intelligence
CGA	conjugate gradient algorithm
CMA	constant modulus algorithm
EVM	error vector magnitude
GoB	Grid of Beams
IoT	Internet of Things
Isomap	isometric feature mapping
LCMV	linearly constrained minimum variance
LLE	locally linear embedding
LMS	least mean square
LS-CMA	least square constant modulus algorithm
LTE	Long Term Evolution
MDS	multidimensional scaling
MIMO	multiple input and multiple output
mmWave	millimeter wave
MVDR	minimum variance distortionless response
PCA	principal component analysis
RLS	recursive least square
RMS	root mean square
RX	receive
SINR	signal interference plus noise ratio
SMI	sample matrix inversion
SS	synchronisation signal
SS-RSRP	synchronisation signal reference signal receive power
SSB	synchronisation signal block
SVD	singular value decomposition
SVM	support vector machine
UE	user equipment
TX	transmit

I_1	intensity value 1
I_2	intensity value 2
L_1	level 1
L_2	level 2
p_1	pressure 1
p_2	pressure 2
$\mathbf{X}$	data matrix
$\mathbf{U}$	orthogonal matrix in SVD
$\mathbf{\Sigma}$	unique matrix in SVD
$\mathbf{V}^T$	orthogonal matrix in SVD

1. INTRODUCTION

Telecommunication technology has expanded incredibly quickly over the past few decades. Today, the technology wave has forced companies to renew and develop different methods of offering faster and more reliable networks. We live in a new era in the telecommunication field, in which 5G has finally come into use. 5G is the successor of 4G Long Term Evolution (LTE) and promises improvements for end-users in communication. In particular, 5G provides increased throughput and a lower network outage probability [1 p. 347], which are highly significant features today, when almost everything is dependent on a network. The network plays a significant role in the different aspects of infrastructure, and many essential aspects rely on the network. Therefore, network outages should not occur, as these might lead to remarkable catastrophes. From the consumer perspective, 5G offers lower battery consumption [1 p. 347]. Practically, it means that mobile industries can develop more efficient phones for customers. Additionally, 5G decreases traffic fees because the maintenance of the network infrastructure does not consume money in the same way as before [1 p. 347].

At the same time, artificial intelligence (AI) has become more popular. AI appears increasingly often in different industries, especially machine learning, a subset of AI. AI and machine learning provide new opportunities for handling a considerable amount of data, which is otherwise difficult for a human to handle by him- or herself. However, AI is overshadowed by the belief that AI and machine learning replace human labour. Machine learning is an aid and it is not intended to be a replacement. Instead, machine learning supports business and makes different industries' fields more productive [2 p. 13]. Machine learning and its roots are based on statistics used to find patterns from a large amount of data. Moreover, it provides several concepts to solve problems, such as classification and prediction. These concepts utilise the similarity of reference data to develop a model for the problem. Creating such a model requires an understanding of the problem and its variables to achieve the most accurate model.

Companies have different organisations and structures, in which machine learning would serve as a handy and practical tool, particularly in testing. Product testing plays a significant role in the radio design and development field. It is imperative to test that a product will fulfill the customers' needs. Therefore, it is essential to design various test cases to assess how the products behave and determine their limits under ordinary and extreme conditions. Testing occurs in many different phases during product development and includes a time-consuming analysis, which might lead to human errors. Automation is one solution for test cases, which are repeatable for the products. For time-consuming analysis, machine learning is the best toolkit for helping to develop an automated testing environment. To create a more effective automated testing environment, the environment must consider different opportunities that machine learning can offer. With suitable solutions and models, machine learning provides effective analysis for testing.

This thesis studies the possibility of utilising machine learning in an automated test environment for analysing emerging samples and to achieve an efficient radio production pipeline. In addition, it is studied how well a random forest classifier performs at classifying beamforming samples. Moreover, the random forest classifier is compared against the alternative method, which is not utilising machine learning.

The hypothesis of the research question is that machine learning is a more suitable solution to classify beamforming samples and the random forest classifier offers an adequate solution with high accuracy. In this thesis, we highlight several features to define beamforming in a manner simpler for analysis, and the final model is integrated into the application to evaluate newly emerging beamforming samples.

1.1. Scope of Master's Thesis

Visualisation provides a better opportunity to understand data. However, manual interpretation and analysis might take time, even for specialists. Radio includes many components and various architecture solutions to serve telecommunication. Moreover, it contains several different signal processing techniques based on advanced mathematics and many complex algorithms. Therefore, this thesis concentrates on beamforming only, which is a signal processing technique used for wireless telecommunication.

This thesis focusses on building an analysis system for radios, using machine learning methods and algorithms to handle data and learn the patterns of existing beamforming samples. Beamforming is a broad notion and contains many beamforming types with different architecture solutions. Although beamforming employs many complicated mathematical calculations and has various types of techniques, we introduce the minimum criteria for beamforming analysis to demonstrate that beamforming has succeeded on the radio.

This thesis consists of seven chapters. The subsequent chapter contains the literature review, which briefly explains the history of telecommunication technology. Moreover, the chapter explains what beamforming is and how today's beamforming works in 5G. The chapter contains a literature review of machine learning, as well. It introduces the most popular machine learning methods and provides detailed information about the topics that have been used in the implementation. At the end of the chapter, we then explain how machine learning has been used in the telecommunication field. The third chapter, *Three Main Features in Beamforming*, introduces several features for simplifying beamforming analysis and explains the relationships between features and beamforming. After the theory is presented, the fourth chapter introduces the invention of the thesis and the phases of the implementation. Following this, the fifth chapter presents the results and discusses how well the implementation works in practice. The results are compared with the alternative method to present the performance of the implemented model. Afterwards, the sixth chapter presents the achieved results and the conclusions that can be drawn based on these results, and it introduces ideas for alternative solutions or approaches regarding how the current work could be improved. Finally, the seventh chapter summarises the work and presents the conclusions.

1.2. Contribution

The final model of this thesis is part of a major company's automated testing environment for radios. The model analyses beamforming from radio and categorises the beamforming results into passed or failed classes, which serve as feedback for a radio architect. The motivation for the research is that the analysis phase should be automated for the testing environment, which validates either a final product for customers or a returned faulty product not working as expected. The high-quality testing removes unnecessary steps in the deployment. Radio installation to an antenna tower is not inexpensive or safe work; it includes many risks, and therefore with high-quality testing, the related risks can be mitigated and expenses reduced. Therefore, there is a need to develop an automated analysis system to remove unnecessary installation steps and ensure that radios are working correctly from the beginning.

This thesis does not include its own data gathering process but instead uses data that has been collected from an automated test environment. The data provides necessary variables to classify beamforming on radio; these variables are explained in the third chapter, which presents the three main features in beamforming to simplify the analysis. To develop an automated analysis system with a supervised machine algorithm, we needed to perform some background work to gather training material. In this regard, we needed to classify samples manually for the learning process. Manual classification is time-consuming, and human error might occur during the labelling. Therefore, automated analysis is needed for the test environment to accelerate the analysis and eliminate human errors.

The contribution of this thesis for the testing environment is to utilise gathered samples from radio, which contains success and failure cases, to build a machine learning model. The model will solve the beamforming classification problem and classify the results as feedback for the end-user by utilising the random forest classifier. The model has been stored in an application that provides feedback for currently the one radio variant.

2. LITERATURE REVIEW

This chapter introduces two different technology areas and unites its primary findings. This thesis contains two state-of-the-art sections, and this chapter briefly explains the significant main entities regarding telecommunication and machine learning. Furthermore, this chapter describes the most influential aspects of radio evolution, which have led to 5G and beamforming. Finally, this chapter presents the most essential topics related to machine learning, particularly the subjects, which belong under the classification.

2.1. History of Radio

Communication is a necessary skill that helps to ensure humanity's survival, and it makes it possible to send complex messages between humans. The communication format has been developed for centuries, and the distance that messages can be sent has quickened the communication evolution. From smoke signals and pigeon posts to the communication of today, reliability and speed have increased significantly. Telecommunication has made globalisation possible, and news and data can reach others in seconds, even from the other side of the world. Telecommunication is based on fundamental communication, in which the message delay is sought to be as minimal as possible. Telecommunication has many significant milestones in history, such as telegraphs, telephones, wireless telegraphs, radios, and televisions.

The radio is a communication invention that uses electromagnetic waves in communication. The frequency range of non-ionising electromagnetic waves is 3 kHz to 300 GHz [3 p. 669]. This range contains two separate electromagnetic waves: radio and microwaves. Radio waves include a frequency range from 3 kHz to 300 MHz, while microwaves have a frequency range from 300 MHz to 300 GHz [3 p. 669]. Both types of waves are utilised to transfer data for several different purposes. That it is possible to utilise the aforementioned electromagnetic waves, the radio must contain converters, a transmitter, and a receiver, which modify data for transmission. During transmission, the transmitter converts data to radio waves, and when the other radio receives the radio waves, the waves are converted back to their original form by receivers [4].

Today, radio communication offers many different types of techniques that allow for the transfer of data, called modulation. One of the most common radios is AM and FM radios, which utilise amplitude or frequency in the modulation process. The AM radio was one of the first inventions in radio technology, in which amplitude modulation has made it possible to use many radio stations [5].

In telecommunication evolution, when mobile phones entered the market, there was a need for improved wireless communication. Therefore, wireless communication was studied more, and new techniques were learned to improve communication. Beamforming is one of these techniques and is utilised in today's radio to provide efficient and reliable wireless communication for users simultaneously. Today, better solutions are needed to serve more intelligent infrastructures and avoid data losses and transmission delays. Therefore, beamforming is studied more to improve performance and transfer data more quickly than before. Notably, there is a great need to transfer the considerable amount of data present today. The other motivation to improve beamforming is that there is a significant need to work with different data types, which demands reliable communication.

2.2. Effect of Beamforming in Telecommunication Technology

Beamforming is a signal processing technique in wireless telecommunication used in antennas' sensor arrays for directional transmission and reception [6 p. 4]. The beamforming technique creates a focussed signal for a receiver device from a transmitter source to provide an intense connection. It is the mixture of radio signals from non-directional antennas, which occurs by steering the antenna arrays electronically. Beamforming aims to reduce interference and improve communication capacity between antennas. In beamforming, the amplitude and phase in each antenna element are controlled so that it is possible to form and oversee the main lobe of the beam and the related side lobes to maximise the gains and minimise interference [7 p. 1].

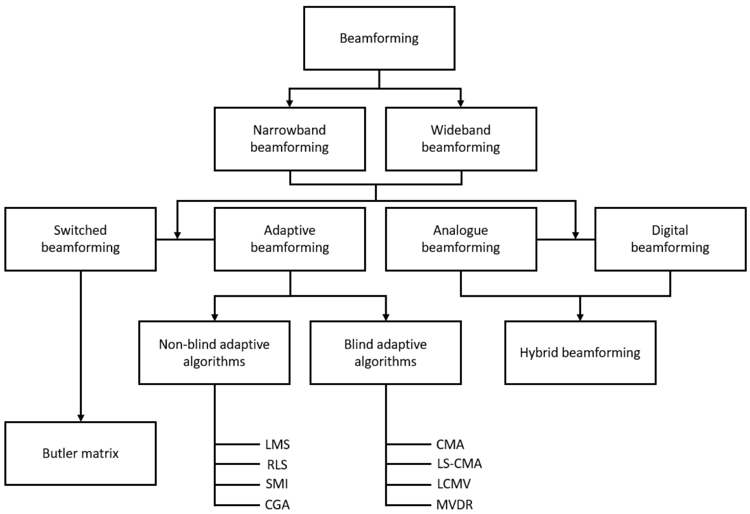

Figure 1. One method of classifying beamforming into several subsets. The division of the beamforming occurs based on different architecture solutions to compute the beams.

Although beamforming sounds as though it is a new trend word in telecommunication technology, especially in 5G, beamforming principles were invented in the 1940s [8]. Beamforming contains a large amount of complex mathematics; therefore, it is difficult to analyse beamforming, particularly in determining why communication between antennas causes communication failure. In addition, beamforming has different variations. Thus, it is possible to categorise beamforming types into classes and subclasses [9 p. 759]. Figure 1 presents one method of classifying beamforming. The mathematics and antenna architecture behind the beamforming variation of classes is different from each other. Therefore, it is not straightforward to develop a general analysis system for all beamforming types.

However, this thesis highlights several features that can be utilised in a different type of beamforming, analysing beamforming success. Detailed information on these features is presented in the chapter *Three Main Features in Beamforming (Chapter 3.)*.

Digital beamforming is not the newest invention, and it has been studied over the past century. It is based on a need for radio techniques to have better performance in communication. Digital beamforming is a technique that can be used on the transmitter side to send information or on the receiver side to receive information. It provides many advantages in communication, particularly on the receiver side. The benefits include improved adaptive pattern nulling, better resolution, multiple beams, antenna self-calibration and much smaller side lobes, array element pattern correction, flexible radar power, and time management [10 p. 50]. Digital beamforming provides more flexible transmission for data than analogue beamforming because digital architecture supports as many RF chains as antenna elements [11 p. 7]. The architectural difference between digital and analogue beamforming in the antenna element is illustrated in Figure 2.

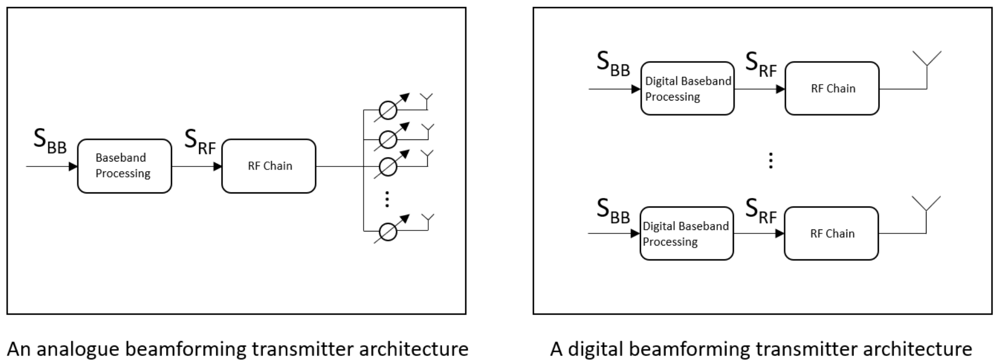

Figure 2. Architectural differences between analogue and digital beamforming.

However, digital beamforming is not always the best solution for every device in practice, especially 5G devices. The reason is that hardware requirements and complexity might significantly increase the energy consumption and cost. This might cause difficulties with integrating a digital architecture structure into mobile devices to achieve the maximum benefit. Therefore, the best practical use case for digital beamforming is in base stations, in which performance plays a more critical role than mobility [11 p. 8].

2.2.1. 5G and Beamforming

5G is the next generation of mobile broadband, offering faster download and upload speeds. 5G differs slightly from its predecessor, 4G LTE. For example, 5G offers three different spectrum stages, unlike 4G LTE, and these different stages provide separate coverage areas for communication. A considerable difference between 5G NR and 4G LTE is that LTE cannot use millimeter waves (mmWaves) for data deployment. In the high-band spectrum, mmWave technology can offer the best performance for 5G. Using mmWaves can reach 5 Gbps speed with very low latency, which significantly speeds up and stabilises data transmission [12 p. 1–11]. The downside to using mmWaves is that they are only used over small coverage areas, and building penetration is poor for small radio signals [13 p. 80454].

As previously mentioned, 5G utilises spectrum ranges to provide widespread coverage and supports different use cases, which rely on the network. The low-band spectrum works with frequencies smaller than 1 GHz and offers broad coverage [14 p. 6]. Broad coverage is a suitable solution for urban, suburban, and rural areas. Its significant advantage is the ability to better penetrate walls by utilising its small length of waves [14 p. 6].

The opposite of the low-band spectrum is the high-band spectrum, which works with frequencies larger than 6 GHz. The advantage of the high-band spectrum is that it provides ultra-high-speed mobile broadband in 5G. However, at the same time, it does not offer as proficient wall penetration as does the low-band [14 p. 6]. Therefore, a mid-band exists to solve the differences between the previously mentioned two spectrums. A mixture of the advantages offered by these two spectrum ranges, the mid-band offers a combination of coverage and capacity for 5G service and works between the 1 GHz and 6 GHz spectrum range [14 p. 6].

In different spectrum areas, beamforming is utilised to provide better communication. Beamforming is necessary for 5G, particularly in mmWaves frequency-based transmission, because the technique reduces the complex structure of the hardware and the consumption of energy [15 p. 1].

2.2.2. Relationship between Beamforming and Massive MIMO

Multiple input and multiple output (MIMO) is a system that consists of multiple antennas at transmitter and receiver radios [16 p. 124]. MIMO is a legacy of LTE for 5G, and massive MIMO is the advanced form of MIMO, using more than 32 logical antenna ports to communicate in 5G [17]. Massive MIMO offers better capacity and coverage for radio software, thereby providing, in practice, more efficient communication outdoors and especially indoors [18].

Let us simplify what happens in massive MIMO in practice and go through a 4x4 MIMO example. Figure 3 illustrates the data split into four different sections for transmitters. The transmitters transfer data sections parallel to the receivers, which decode the received signal back to its original information. The advantage of this technique is that data streams increase the data rate, resulting in faster wireless communication [19 p. 24]. The benefit on the receiver side is that MIMO provides receive (RX) diversity, which improves sensitivity in receiver antennas using polarities and multiple radio channels to transfer data [20 p. 4].

Both MIMO and beamforming are techniques used to increase capacity, but these methods have different architecture to perform more efficient data transfer [20 p. 17]. In mmWave frequencies, massive MIMO requires the beamforming technique to take advantage of large bandwidth. Building an efficient wireless connection that can simultaneously decrease interference and minimise data loss requires massive MIMO and beamforming techniques to prevent a too-large energy consumption. MIMO antennas require D/A converter circuits to convert digital signals to analogue signals

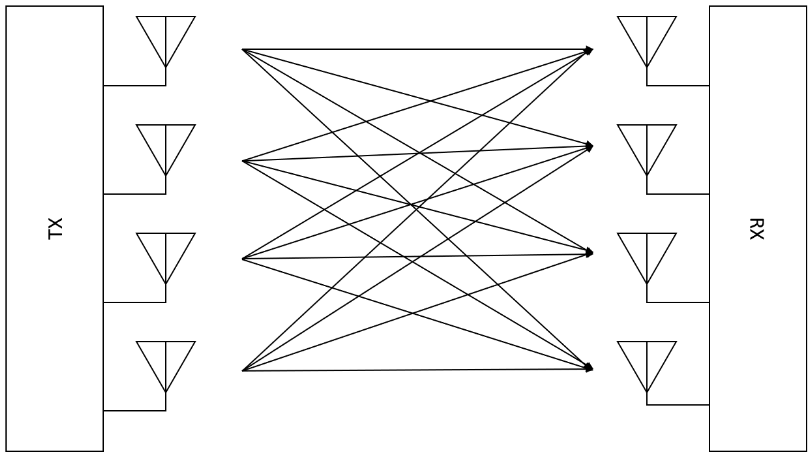

Figure 3. Illustration of a 4x4 MIMO—how to transmit (TX) device split data into four separate sections and transfer information parallel to RX device.

that antennas use to can transfer data [21]. If mmWaves frequencies are used in digital beamforming, the antennas will require a high-speed D/A circuit, and therefore, operation leads to significant power consumption [21]. This scenario should be avoided as often as possible, and the solution to this problem is to utilise hybrid beamforming. Hybrid beamforming is not a new invention and was invented more than 10 years ago, but during the past years, the advantages of the techniques have been noticed in massive MIMO [22 p. 10]. The method reduces the power consumption because part of the signal processing is conducted in analogue antenna elements. In addition, this technique frees up calculation effort for other processes, and therefore, hybrid beamforming is the recommended technique for mmWaves with massive MIMO [21] [22 p. 10].

2.2.3. Grid of Beams

A different type of radio architecture exists to provide high spectral efficiency and coverage for users. This thesis uses data from the radio, which utilises the grid of beams RF architecture.

Grid of beams (GoB) is one key component alongside a massive MIMO antenna array to provide better connectivity in 5G. GoB solves some problems on the massive MIMO side. Although massive MIMO is an efficient technique, GoB provides fixed wideband beams when user equipment (UE) sees only effective channels [23 p. 1]. Moreover, GoB offers lower computation complexity, while massive MIMO demands significant computing capacity when channel matrices increase to be too large [23 p. 1].

In 5G, the coverage is beam-based, indicating that in practice, a radio provides beam patterns that construct statics or semi-statics synchronisation signal block (SSB) beams [17]. The specified size of the cell might then contain many separated beams to cover the cell area. Although 5G and beamforming are discussed often, 5G does not always need beamforming. In such cases, the entire cell area uses only one beam to cover the

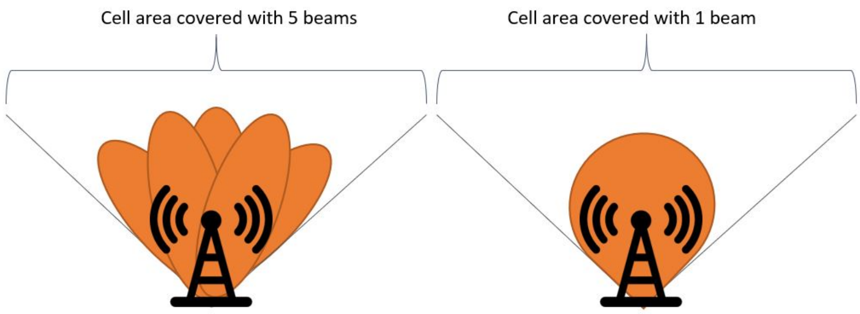

Figure 4. Illustration of the differences when a pack of the beams covers the same area rather than only one beam.

area [17]. Figure 4 presents differences between how well multiple beams can cover an area versus only a single beam.

2.3. Overview of Machine Learning

Today, AI and its various subfields are discussed everywhere. Sometimes, the meaning and the content of AI and its subfields are mixed up, primarily when discussing machine learning. It is good to know the differences of each group so that it is easier to understand the purposes of the categories.

The brief and straightforward definition of AI is that people aim to imitate human intelligence for a computer such that the computer can perform cognitive processes [24 p. 8]. AI is the highest level of the hierarchy and can be divided into many subcategories. Therefore, the categories can be visualised as in Figure 5, which presents one way to visualise the structure of AI and its subfields and categories [25]. The second central part of this thesis is machine learning, and this chapter concentrates on explaining its significant elements.

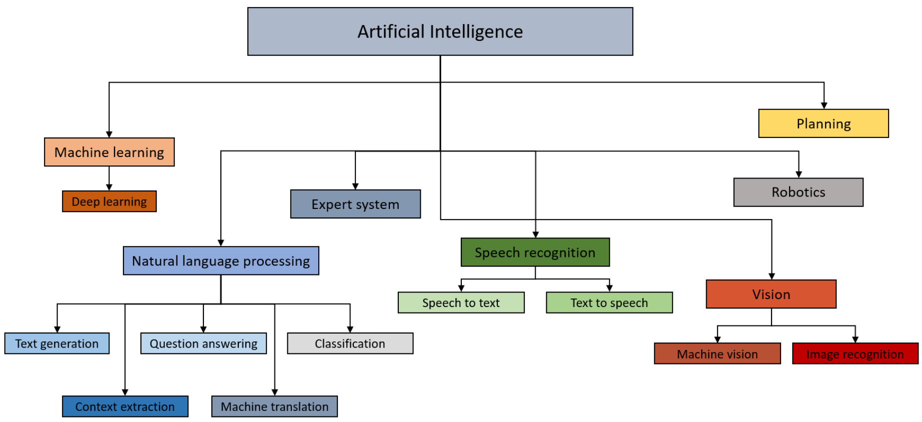

Figure 5. One method of visualising AI and its subfields and applications.

The definition of machine learning is that computers can learn something from data. Data works as an input for a computer, from which the computer aims to learn the data behaviour instead of understanding how the information is explicitly programmed [26 p. 1–3]. The machine learning field concentrates on creating algorithms through which computers can process data and find patterns from it. Depending on the purpose of the application, the information from data can be utilised to make classifications or predictions [26 p. 8].

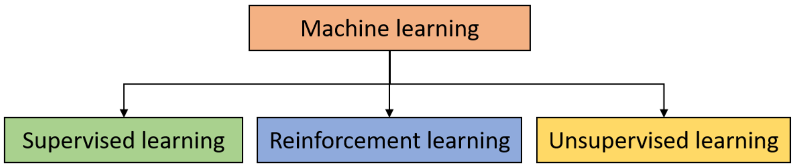

Figure 6. Categories of machine learning.

Machine learning is based on creating and using algorithms to allow a computer to learn automatically. Algorithms can be divided into three main categories, as presented in Figure 6. The main classes of algorithms are supervised learning, unsupervised learning, and reinforcement learning. However, one additional category exists that is not always viewed as a main category. Semi-supervised machine learning is another broad aspect of machine learning, and its algorithms are a mixture of supervised and unsupervised learning [27 p. 3].

Supervised learning is an algorithm class in machine learning that uses labelled data to create a regression or a classification. Instead, unsupervised learning is an algorithm class in which the algorithms do not receive labels. Therefore, unsupervised methods learn models and patterns by observing data because the data does not have already-labelled information, and the conclusions must be drawn based on the observation of the data [28 p. 2]. In contrast, supervised methods learn from examples and find similarities between them [28 p. 2]. The classic examples of unsupervised learning are clustering and dimensionality reduction. Clustering is a task in which the data points are identified and assigned to the clusters to present the instance of the group [26 p. 236], as opposed to dimensionality reduction, which is a technique that can remove noise and data complexity, thereby reducing the dimensionality of the data set. This technique reduces the possibility of overfitting during a learning phase [26 p. 213]. Overfitting is a phenomenon in which learning algorithms attempt to build a too-exact or -complex model based on noisy data, and therefore, an overfitted model might cause false predictions and classifications in supervised learning [29 p. 126–127]. Instead, algorithms should develop models that can be generalised to new unknown examples to solve the problem.

2.4. Supervised Learning

The machine learning problem of this thesis is the beamforming classification. This problem needs a solution to learn correct beam patterns from data and classify newly emerging samples based on their data behaviour. The classification is a part of the automated testing environment to analyse beamforming on the radio and provide the behaviour of the results as feedback.

Supervised learning is one of the main algorithm classes in machine learning, and its algorithms use labelled data to learn patterns and develop a model to generalise new data, utilising learned information about existing features from the data. Supervised learning offers a variety of techniques to solve problems, and it can be divided into two subcategories: regression and classification [26 p. 8][30 p. 66].

Regression is a technique of learning to predict continuous outputs, and it can be applied to study statistical relationships between studied features [31 p. 1]. Furthermore, correlation can be calculated to determine how robust a feature relationship is. When correlation has been found between features, it is possible to create a predictive model based on the features that have a relationship [26 p. 8].

Meanwhile, classification is a classic problem in machine learning, and most of its methods belong to the supervised class. As with regression, the classification method seeks to find relationships from input data. In regression, output values attempt to predict based on input data. In turn, in classification, the model is built to predict discrete classes from unknown samples [26 p. 8] [30 p. 66]. To clarify the differences between regression and classification, Figure 7 presents an example of these two methods.

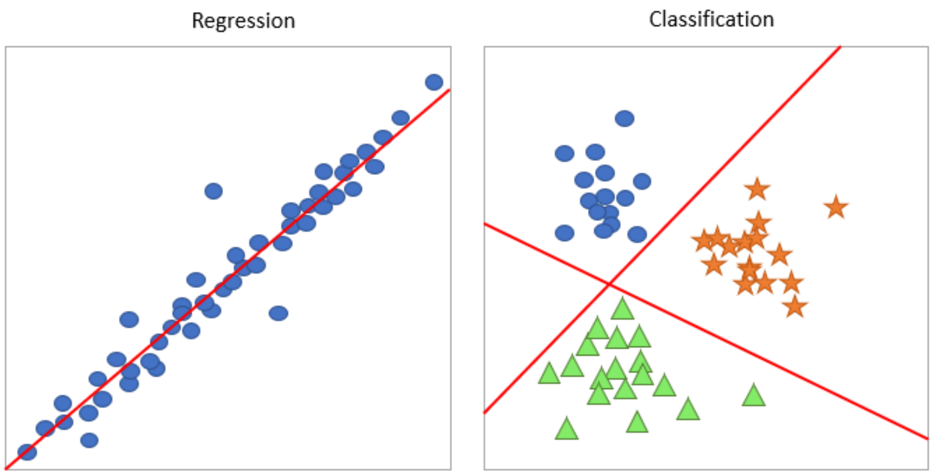

Figure 7. Examples of regression and classification.

2.4.1. Classification Methods

Classification is one of the most common problems in machine learning, especially in intelligent systems. Therefore, many classification methods exist to solve various classification problems. Classification methods can be divided into logic-based algorithms, perceptron-based techniques, statistical learning algorithms, instance-based learning, and support vector machines [32 p. 250–260], and many of them have been implemented into the Scikit-learn library. Scikit-learn is Python's machine learning library, providing many different machine learning methods and functionalities to implement intelligent systems swimmingly. Therefore, the library was used in this thesis to implement the classification model and analysis system, which analyses the beamforming samples.

2.4.2. Random Forest and Its Fundamental Part Decision Tree

This thesis focusses on using the random forest method to solve the classification problem. Random forest is one of the powerful supervised machine learning algorithms today, and its fundamental aspect is the decision tree algorithm [33 p. 5]. Random forest uses multiple simple decision tree models to increase the accuracy of the classification [33 p. 5] [34 p. 316]. Decision tree is a comprehensive classification method that can also be used in the regression. It can handle multioutput tasks, and the technique is good at dealing with complex data sets [26 p. 175]. Decision tree is a hierarchical graph, in which the nodes have weights for the decision to lead to another node, until the lowest node has been reached. Decision tree classification is a recursive process, in which the current achieved node is based on choices from previous nodes [35 p. 165–166]. Figure 8 presents a small example of the classification case and how the trained model would make its decisions.

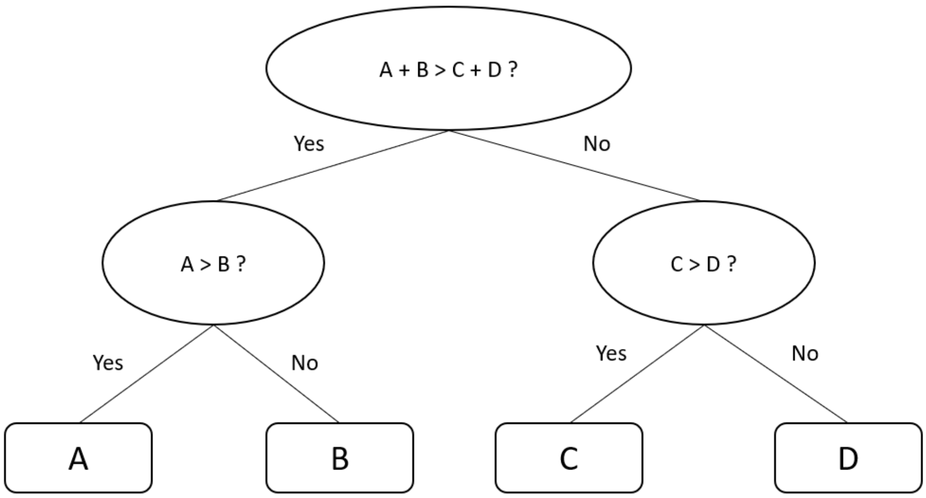

Figure 8. Small and simple illustration of decision tree and how it classifies inputs.

Although decision trees are simple to use and inexpensive to construct, they can handle different data types, such as nominal, numeric, and textual [36 p. 1115]. Moreover, this method offers easy and quick ways of interpreting small-sized trees. However, when data grow large, a decision tree has a few disadvantages. Its disadvantages are that the design time of the tree might grow too large when learning is a laborious, time-consuming process. In addition, too-simple trees do not represent non-rectangular regions well [37 p. 129].

A decision tree might also easily overfit the model, especially when a tree is particularly deep. Overfitting is caused by irrelevant features, which provide unnecessary information and lead to fitting too much to the training instances [38 p. 119]. Practically, this means that the classifier learns too much from the training data and attempts to build an overly complicated model, and therefore, it cannot correctly classify testing data or newly emerging samples. The too-complicated model might cause misclassification with the results in use and decrease the accuracy. Therefore, in the model-building phase, the irrelevant features are aimed to be eliminated before learning by decreasing the data set complexity and lowering the likelihood of causing overfitting in the model [38 p. 119]. The simpler data set makes it possible to generate models for the problems, and therefore, the feature reduction technique is one way to help models avoid overfitting.

The second main problem in decision tree is that it does not always offer the optimal solution. The reason is that a tree concentrates on selecting the best solution in every layer in a tree, but the best choices of the upper level will not always end up yielding the most optimal results at the bottom of the tree [39]. Random forest aims to solve these two problems by using several randomly generated trees to find the most optimal solutions. It can utilise two approaches to achieve the best optimal solution, namely by aiming to reduce error due to bias and variance [39]. Therefore, random forest models are powerful machine learning tools today because they utilise a collection of decision trees whose results are aggregated into the final result [33 p. 5] [39]. The key of this technique is to use many randomly built trees to avoid overfitting and reduce variance to obtain the best possible model without too-large biases.

2.5. Dimensionality Reduction

Sometimes, data may contain many different types of information, called features. Too many features may sometimes be a problem in analysis and make it more difficult to find a good solution. All features might not be necessary or do not provide helpful information to solve a problem. In addition, if data includes many features, the data will need a larger number of samples to study the behaviour of the data set. Furthermore, too many features might lead to overfitting when the classifier aims to fit a model into complex a data set. As the previous section explained, overfitting is a common problem in machine learning and affects a large portion of the modelling results for finding a suitable solution. Overfitting causes the final model to be unusable in practice, because it will not understand newly emerging samples and might provide inappropriate output values for the samples. Therefore, a significant motivation is to perform a dimensionality reduction for the data set to achieve a model that can be generalised to unknown samples.

Although the most significant motivation is to develop a model that generalises data, the dimensionality reduction offers other advantages, as well. For example, it provides better accuracy when misleading and redundant features are removed from the data. Removing features decreases noise in the data, making it possible to fit a simpler model to the data set [40]. In addition, the dimensionality reduction decreases the computational burden to build a model, which, for a training phase, takes less time.

Furthermore, a smaller pact of features allows for adding more samples to the data set when features do not require as much space as earlier [40].

Dimensionality reduction can be split into two approaches: feature selection and feature engineering. In feature selection, the relevant features are identified and selected from the data and are used in further research. In contrast, in feature engineering, data are processed manually and transformed into new feature representation from existing data [40].

Many dimensionality reduction techniques exist today, and the methods can be divided into two categories: linear and non-linear dimensionality reduction methods [40]. Linear dimensionality reduction contains the most common and well-known reduction strategies. Its most popularly used method is principal component analysis (PCA). Other well-known methods include factor analysis and linear discriminant analysis [40]. These methods are often used to reduce data features when the data lie on a linear subspace. In the alternative case, when data does not lie on a linear subspace, nonlinear dimensional reduction methods are used. The most popular methods are multidimensional scaling (MDS), isometric feature mapping (Isomap), and locally linear embedding (LLE) [40]. Although there exist many different methods to reduce features from the data, this thesis concentrates on using PCA in feature reduction.

2.5.1. Principal Component Analysis

Principal component analysis is one of the most popular dimensionality reduction algorithms. It is a good choice when data lie on linear subspace and the importance of the variables cannot be identified directly, but at the same time, there is a great need to simplify the data set and reduce features from it.

PCA has four main goals: to extract the most relevant information from the data set, compress the size of the current data set, simplify the description of the data set, and analyse the structure of the observations and the variables [41 p. 3]. To achieve the aforementioned goals, the PCA method aims to find components and order them by variances, such that the first component preserves the most significant variance [41 p. 4].

Singular value decomposition (SVD) is a method applied in different mathematical areas and is utilised in PCA to find the principal components from the data matrix. This method is a standard matrix factorisation technique, which can decompose the training set matrix into three multiplicable matrices $\mathbf{U}$, $\mathbf{\Sigma}$ and $\boldsymbol{V^T}$ [26 p. 221]. The formulation of SVD is presented in Equation 1, where $\mathbf{X}$ presents the training set matrix, $\mathbf{U}$ and $\boldsymbol{V^T}$ present orthogonal matrices, and $\mathbf{\Sigma}$ is a unique matrix with eigenvalues from the training set matrix. In PCA, $\boldsymbol{V^T}$ is the most interesting matrix, which includes all unit vectors that define all principal components [26 p. 221].

$$X = U\Sigma V^T \tag{1}$$

Although PCA reduces and simplifies features from the data set, the most challenging aspect is selecting the number of principal components for the learning phase without losing too much information from the data set. Therefore, the principal components of the data set should be studied further, and the number of principal components should be selected, including a suitable amount of cumulative variance [26 p. 223]. Instead of arbitrarily choosing the number of principal components, the better option is to take a large portion of variance, which can, for example, include 95% of the total variance [26 p. 223].

In this thesis, we used the 95% cumulative threshold to find all principal components included in the threshold value. Moreover, we studied how the principal components included in the threshold would affect the accuracy and whether it is possible to use fewer components without losing accuracy.

2.6. Machine Learning in Telecommunication Technology

Machine learning has been made possible to develop intelligent applications and programs that learn patterns and behaviours from data. When a relationship between variables has been found, it is possible to make predictions based on existing data. In telecommunication, a large amount of data is created at all times, and therefore, machine learning has begun to be utilised in the telecommunication field to develop more intelligent applications.

Machine learning has been used widely in telecommunication, and it can be utilised in different aspects of a business, such as in customer interfaces, at the management level, and in communication systems. In this thesis, we provide a solution for the communication systems to increase the effectiveness of beamforming analysis. However, this is not the first study to utilise machine learning in communication systems, particularly in its analysis. Machine learning can be utilised in many different areas. For example, one existing study concentrates on analysing CPU loads from the base station using existing machine learning methods to develop a monitoring model that detects abnormal situations [42]. In addition, machine learning has been used already in a test automation environment, and a big data analysis has been conducted and implemented into the system to analyse CPU and memory loads [43]. Nevertheless, machine learning does not depend on the technical field, and it can be utilised in several fields within telecommunication, such as the customer interface. Machine learning is used to create intelligent chatbots, which provide answers to customers' questions and solve their problems. Chatbots increase customer satisfaction when the customer receives service as quickly as possible, and queuing and waiting do not occur in the same way they have previously [44][45]. In addition, machine learning can be used at the management level, in which machine learning serves as a valuable tool to find patterns from data and automate processes and analytics to predict customer value. It helps the management to more efficiently lead the company [45].

Still, machine learning should be used more in the telecommunication field. At the moment, machine learning is used to optimise mobile tower operation and assist in preventative maintenance. Communication service providers have many different concerns with maintaining mobile towers. The maintenance of the mobile towers is time-consuming because inspectors must scrutinise the infrastructure and ensure that every device runs correctly, such as power generations and air conditioners [44]. One of the concerns in mobile tower sites is uninvited guests who wish to steal valuable equipment at towers, and therefore, real-time video analysis systems have been built that can analyse the entire environment through a 360-degree camera. Furthermore, different Internet of Things (IoT) sensors can be installed to respond to changes in the environment and abnormal alert situations using machine learning algorithms [44]. An abnormal situation can be a fire, which the sensor can recognise by utilising changing parameters.

As we can see, machine learning can be used in many different types of problems and has increasing effectiveness in business. In this thesis, we wish to increase the effective analysis for beamforming. This master's thesis is part of the radio test automated environment and focusses on studying whether supervised learning methods can be used to solve a beamforming classification problem, particularly utilising the random forest classifier.

3. THREE MAIN FEATURES IN BEAMFORMING

As demonstrated in the previous chapter, beamforming includes many variations of architectural solutions to offer better data transmission, and these variations have many complicated mathematical processes working behind the devices to handle wireless communication. This chapter introduces the main aspects of beamforming that contribute to the results. It is good to know a few essential elements and their meanings when building the beamforming analysis model. The elements simplify the beamforming and allow for gathering meaningful features for the learning process. Although beamforming contains much complex mathematics working behind the antenna elements, it is possible to identify a few features that serve as successful beamforming criteria. The three main criteria for successful beamforming are the similarity of the polarities, the pattern of beam set, and the throughput.

3.1. Role of Polarities in Beamforming

Polarisation is part of the antenna port, where the radiators generate a wave that moves along the same plane [46 p. 17]. Three types of polarisation exist: vertical and horizontal linear polarisation, slant polarisation, and circular polarisation. Slant polarisation is generally set at +45 and -45 degrees [46 p. 17].

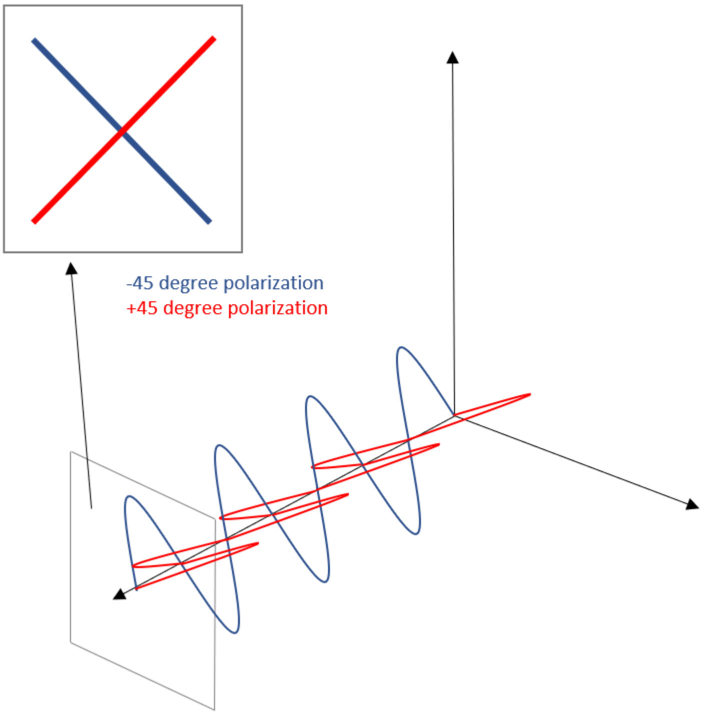

Figure 9. Illustration of orthogonal dual-polarised waves, where angles have been set at +45 and -45 degrees. This configuration is also known as slant polarisation.

Mobile communication employs dual-polarised antennas in the base station to improve performance. In this thesis, we concentrated on collecting data from orthogonal dual-polarised antennas. Orthogonal dual-polarised antennas are typically required in the base station that the same coverage has reached for both polarisation [47 p. 79]. Figure 9 illustrates the movements of slant polarisation, which are set at +45 and -45 degrees. The data is transferred in parallel at the same time using the polarities.

As mentioned previously, base stations typically require the same coverage for both types of polarisation. Therefore, one criterion for successful beamforming is that the polarities send almost the same amount of power from transmitter antennas to receiver antennas. In 5G NR, synchronisation-signal (SS) is used for measurements [48], and in the polarisation comparison, we are interested in the power that a reference signal sends to receiver antennas. In this thesis, synchronisation-signal-reference-signal-received-power (SS-RSRP) measurements are features in the data that were used in the machine learning approach.

Although both polarities aim to send the same amount of power, it is impossible to obtain the same SS-RSRP measurements for polarities, especially in noisy environments with many interference sources. At first, we set a limit for the differences between polarities, and the limit was based on the 3 dB rule. The dB unit is based on base-10 logarithm, where Equation 2 presents the difference of levels, which can be acoustical energy, intensity or power, electrical energy or power, optical luminance, or the dose of ionising radiation [49 p. 10–12].

$$L_2 - L_1 = 10 \times \log[(I_2/I_1)] \tag{2}$$

Intensity is proportional to the square of root mean square (RMS) pressure, so Equation 2 can be represented in the form of Equation 3 [49 p. 10–12]:

$$L_2 - L_1 = 10 \times \log[(p_2/p_1)^2] \tag{3}$$

which can be simplified to the form of Equation 4 [49 p. 10–12]:

$$L_2 - L_1 = 20 \times \log[(p_2/p_1)] \tag{4}$$

For example, if we have two intensity values, I_1 = 1 and I_2 = 2, their relation is 2. This means that $\log(2) \approx 0.3$ dB. When the values are placed in Equation 2, we obtain $10 \times \log(2) \approx 3$ dB [49 p. 10–12]. The value $\pm$ 3 dB works as the double/half point in the comparison, due to which the power ratio of the polarisation is kept smaller than 3 dBm; thus, it can be said that beamforming has been successful on the radio.

3.2. Pattern of Beam Set

Beamforming is a data transmission technique in which the data receiver target has first been located, after which the power is concentrated on the receiver—for example, on the UE—to enhance transmission. One of the problematic aspects of analysing beamforming on the radio is that many beam sets exist, and the analysis should be provided for all possible scenarios. A beam set is a set that includes a specific number of beams that have a directed position. Therefore, it is time-consuming to check during a test phase that the beams are working correctly and point towards the correct place. The pattern of the beams constructs the area, which can be 120-degree and 90-degree regions. In the analysis, the correct size of the patterns should be checked to ensure that the pattern will reach its expected edges but, at the same time, will not spread from over the expected area. Figure 10 presents the main idea in the pattern of the beam sets.

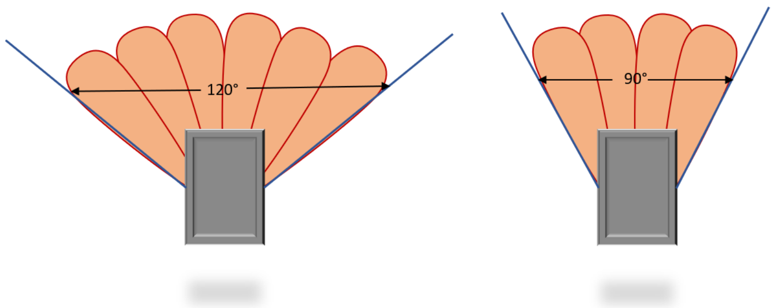

Figure 10. Illustration of the 120-degree and 90-degree regions in beamforming.

This second criterion is based on the invented concept of providing a simplified base to analyse beamforming. The beam set pattern works as one criterion, as it is possible to visualise and understand what should be expected from beamforming.

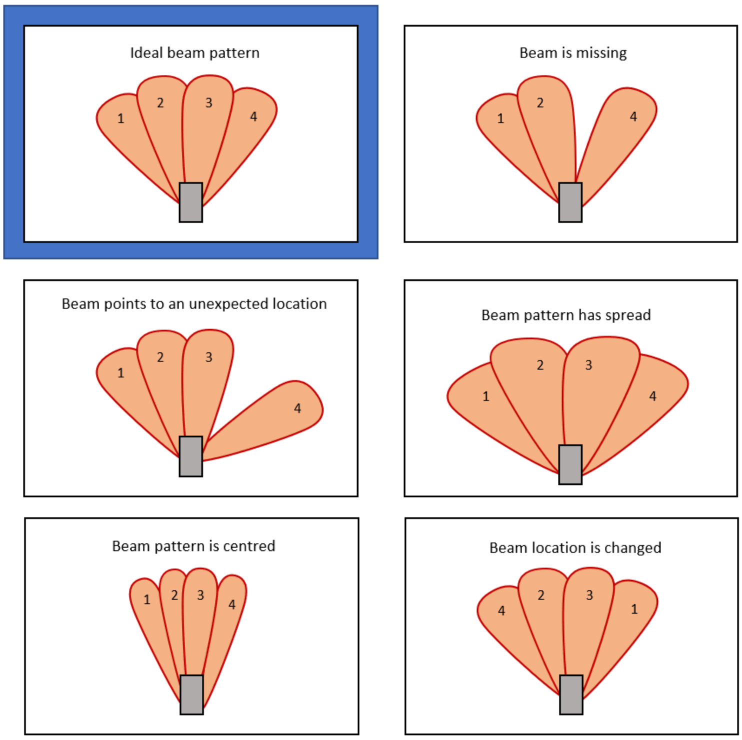

Figure 11. Illustrative figure of the error scenarios in beamforming.

Because beamforming contains much complex mathematics, it might cause error cases if something were to go wrong during the calculation. There can be a case in which the beam does not appear and serve UE, thus causing a disconnection, or the beam has an unexpected location and will not operate at the expected location, thus causing disconnection in the transmission. Another error case is that the beam is spread over the expected area and then serves the customer with too-low power or power centred at an unexpected location, and the coverage area is too small to serve UE. In addition, one of the worst cases is that beams change places among one another. Human eyes do not notice errors in visualised images, but it is an error if the beam is not in its expected location. This might not cause a problem with serving UE, but it might burden the radio if the calculation error were to appear on the radio side. The previously explained errors are depicted in Figure 11.

3.3. Throughput

The final feature of beamforming is throughput. This is an essential measure in wireless communication that indicates how much data has been transferred from a source at a specified time [50][51 p. 159]. Throughput has a significant relationship with bandwidth, which again means theoretical capacity, namely how much data could be transferred from a source at a specified time [50]. Throughput affects the latency, and therefore high throughput measures are desirable because an effective and reliable network keeps customers satisfied. In addition, low latency provides a new opportunity for the low-latency invention, dependent on real-time.

The throughput can be used to determine the success of beamforming on the radio because it is the highest-level measurement, and more detailed measurements directly affect the throughput. However, it is good to be aware of alternative measurements such as the error vector magnitude and signal-interference-plus-noise-ratio (SINR). These are not used directly in this work, but these measurements contain more information to improve the model if they are available.

The error vector magnitude (EVM) measures the performance in the transmitter or receiver by calculating the vector between an ideal reference signal and the actual transmitted signal at a given moment in time. The EVM can measure the amplitude and phase error from modulated signals and achieve the best possible signal performance, which should be as small as possible. In addition, the EVM is a measurement defined as the ratio of the RMS of the error vector average power to the RMS of the desired reference signal power [52 p. 100].

Wireless communication can be affected by several phenomena, thereby weakening the connection. The factors affecting wireless communication performance are propagation loss, fadings, noise, and interference [53 p. 465]. SINR measures the ratio of the desired energy to interference and noise energy, where the rate represents the signal [51 p. 158].

Although there are many beamforming measurements, the analysis can be simplified using throughput. This thesis focusses on simplifying the beamforming scenario and reducing the data to the three main criteria, but the alternative method is to utilise EVM and SINR if the motivation is to find the reason for the failure.

4. IMPLEMENTATION OF BEAM PATTERN ANALYSIS MODEL

The implementation of this thesis is part of a larger test automation environment, and the test environment team has lent support in completing the thesis. The implementation aims to merge two technical fields, namely radio technology and machine learning. Because these two fields are such vast entities, the main elements, which belong to the implementation phase, have only been reviewed in the literature chapter.

The test environment was built to test the performance of the existing and new radio versions. The current test environment can obtain numerous information sources from the radio, but it lacks the automated analysis of new beamforming samples. The implementation aims to provide a suitable analysis model for beamforming to decrease time-consuming analysis processes and intensify the test automation environment.

The research question in this thesis was to determine whether it is possible to utilise machine learning methods to classify beamforming samples. In this thesis, the beamforming test results were classified into passed and failed classes. Therefore, a supervised learning approach was selected for the problem. Accordingly, we used the existing classification method, random forest, one of the most popular classification algorithms today.

This thesis involved several stages, from the data gathering to the final model evaluation, and Figure 12 depicts the main steps of the project sequentially. This chapter presents a detailed description of the intermediate stages of the project and, in particular, the implementation of the machine learning model of this work.

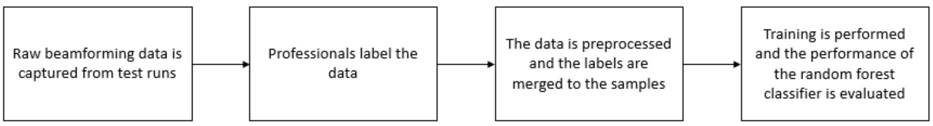

Figure 12. Visualisation of the model development process, which contains all stages of this thesis at high-level.

4.1. Alternative Beamforming Analysis without Machine Learning

Before the existing data for a machine learning approach was sufficient, individual test samples and their structures were studied. At the same time, beamforming professionals analysed samples by sight, which was time-consuming work. The professionals used repeated features to identify the success of the samples and classified them into passed and failed classes. The first feature was SS-RSRP values, which were compared between positive and negative polarities. The goal was to avoid a difference between the values in excess of 3 dBm in the same location. The second feature was to check the angle of the beam pattern, in which the pattern covers the area with an expected angle reaching the expected edges. Throughput measurements were not available but would work as a simplified feature in beamforming analysis. The detailed information is explained in the previous chapter, *Three Main Features in Beamforming*.

At first, the data was not sufficient to work with machine learning methods. Therefore, an alternative method was implemented to analyse individual beamforming samples. This method used SS-RSRP values and their differences between polarities to visualise the results as a green wall. The differences between polarities were marked as black boxes in the green wall images, as in Figure 13. In addition, the alternative method could utilise the second criterion to analyse samples. The method constructed a beam pattern from the numerical data and checked the edges from the pattern to determine whether the expected edge angles were reached. Figure 13 reveals that the samples did not reach the right-side edge, and the error is visualised as a blue column in the figure.

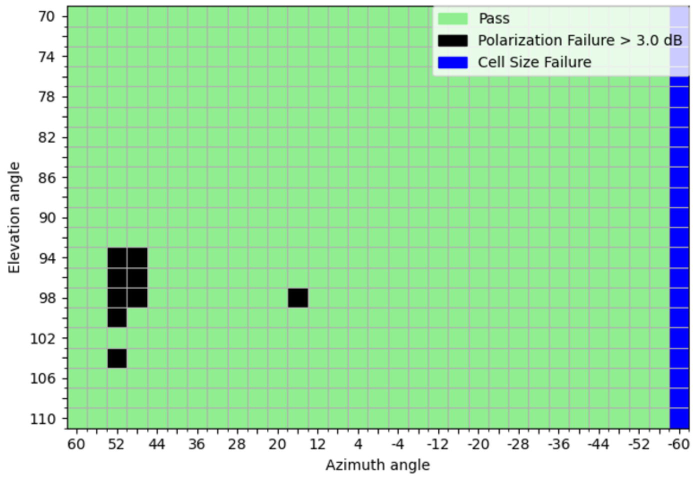

Figure 13. The alternative method detects differences between positive and negative polarities and checks whether the expected angles have been reached. The example contains individual polarity errors, in which the differences between positive and negative polarities are greater than 3 dBm. In addition, the right-side edge was not reached in the beam pattern.

Figure 13 contains one of the samples from the data set for the thesis work. The sample belongs to the failed class, but the greatest reason for this classification does not occur in the image, although the image has some differences with polarities, and the right side of the beam pattern is weak. The main reason is that the entire beam pattern is pointed towards the wrong location. All working samples should have a 0-degree tilting angle in the training material, and the pattern should point perpendicular to the wall. However, the beam pattern was directed to be too low, and the alternative method was unable to notice this, because it does not take a stand of the beam pattern location. This wrong location is one of the errors presented in Figure 11, but the entire beam set has an unexpected location in this scenario instead of one beam. Therefore, it is desirable to improve the current implementation, and machine learning should be utilised if it could solve the problem.

Although one reason for utilising machine learning is to improve the alternative method to find the wrong location, another reason is that the alternative method can sometimes be too stiff and strict to allow for analysing the beamforming performance. The alternative method is based on fixed thresholds, and therefore, it cannot be utilised directly to classify samples when the method finds differences between polarities. It is not easy to achieve a total undisturbed environment. Accordingly, the goal is to decrease the interference in beamforming to achieve coverage areas with good performance for the users. Therefore, a small amount of interference is acceptable, and the limitations can be unnecessary on a certain scale. In this thesis, the goal was to investigate whether machine learning is a more suitable solution for classifying the samples automatically.

4.2. Test Environment

The data was collected by a major company's employees working in the automated testing site. For the thesis work, the data was collected from the test environment, which contained many antennas, set to resemble a curvy wall. The test environment antennas had a rotation ability, and as such, the test environment was capable of collecting more accurate measurements from the beam patterns. The antennas were connected to a spectrum analyser via cables, gathering measurements from the wall. The beam pattern results were swept from the antennas into JSON-formatted data. The raw data served as a data source to train the final model, which was integrated into the analysis application. The analysis application is an extension of the test environment system, which offers beamforming analysis results as feedback to the end-user.

4.3. Program Language and Libraries

For the thesis work implementation, the programming language Python [54] was selected. One of the reasons for this language choice was that the developed model was more comfortable with being integrated into the existing test environment. Nevertheless, the decisive factor was that Python includes many libraries for machine learning; here, the most important libraries are Scikit-learn, Pandas, and Matplotlib for this thesis.

4.3.1. Pandas Library

Pandas is one of the most powerful open-source data analysis libraries and has been created to work quickly, easily, and expressively with data. It supports many different formats of data [55 p. 4–5] and works, among other things, with tabular data with heterogeneously typed columns, such as SQL-table or Excel spreadsheets. Moreover, Pandas works with time-series data, which can be an ordered or unordered arbitrary matrix data. In addition, the library works with any other forms of observational/statistical data sets [56]. Pandas offers many different functionalities for handling data, such as dealing with missing data in data sets, separating data frames into other data frames, and reshaping data sets [56].

In this thesis, the Pandas library played a significant role during the preprocessing phase. In particular, it was used to process raw data from a JSON-format file into data frames to better understand the variables inside the data. The data of this thesis was easier to handle in a data frame format, and the preprocessing was more comfortable using the Pandas library.

4.3.2. Scikit-Learn Library

Scikit-learn, also known as sklearn, is a machine learning library for Python used to build models by employing basic machine learning methods, such as support vector machines (SVM), decision trees, and Bayesian methods [57]. Scikit-learn provides tools for supervised and unsupervised learning but also contains tools for cross-validation, toy data sets to learn how to use the new library, and tools for feature extraction [58]. The library does not have its own data manipulation functionalities, but it works well with Pandas and NumPy, which are Python's package for scientific computing.

In this thesis, Scikit-learn was used to develop a machine learning model using its random forest classifier. In addition, the library was used to perform preprocessing phases, such as dividing the data into training, validation, and testing sets and evaluating the performance of the final model.

4.3.3. Matplotlib and Seaborn Libraries

The best way to understand data is to visualise it. For people, figures are more informative than quantitative data. Therefore, data should always be visualised when possible, as it is easier to detect dependencies between variables from images than directly from numerical values.

Matplotlib is Python's library that provides tools to visualise statistical figures and animated and interactive images for illustration. The library is primarily used for 2D visualisation, but Matplotlib supports some simple 3D visualisation tools, as well [59]. Moreover, Seaborn is another data visualisation library based on the Matplotlib library and provides a high-level interface for drawing attractive and informative statistical figures and tables [60]. In this thesis, the Matplotlib and Seaborn libraries were used to visualise data to better understand principal component analysis and the behaviour of the final model.

4.4. Data Preprocessing and Data Set Description

For the machine learning experiment, the test environment produced 74 beamforming samples using the same radio type and beam pattern. The beam pattern included six beams. In addition, the samples contained a variety of information about the executed test runs. However, only SS-RSRP values are of interest in the analysis. Unfortunately, the throughput values were unavailable for the experiment, though they would have provided valuable information about the performance.

In the raw data, one sample contained the separated measurements from both polarities, positive and negative. Relying on the assumption of the polarities and their mechanism, presented in the chapter *Three Main Features in Beamforming*, the measurements of the polarities can be separated into their respective samples with their labels. Therefore, the total number of samples in the final data set contained 148 separated samples.

After the polarity separation, the data set still needed to be preprocessed into a format that would allow machine learning methods to be used. Still, the data set contained much information. The data structure needed to be vectorised for the learning phase because the raw data included a complicated structure about all measurements from the individual beam. The current raw data can be seen as six separated images, as in Figure 14. Every beam consists of 651 pixels, which are the locations of the sensors in the test environment. Furthermore, the image contained 21 rows and 31 columns, which can be seen as an image of a 21 x 31 shape. All features were SS-RSRP values from the beams, akin to pixels in images.

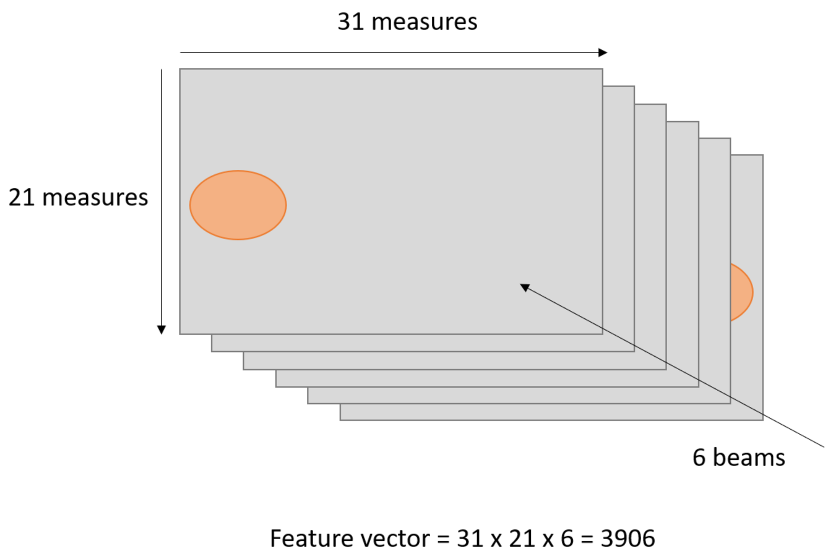

Figure 14. The raw data can be seen as six separate images, where every image includes a specific beam from the beam pattern.

The complicated structure was solved by parsing the data into a long data vector, which contained all the data from the six beams. Therefore, the parsed data included 3,906 features, and the data dimensionality was large.

To execute the learning phase, labels were needed, and therefore, it was necessary that the labels be added manually for all samples using the separated file. The file contained information about the sample ID and the polarity key to merge the labels and the samples.

Table 1. Short summary of the data set

Description of the data set	
The total number of samples in the data set	74
The total number of samples after polarity division	148
Samples with passed label	59
Samples with failed label	89
The percentage of the training set from total samples	80%
Samples in the training set	118
The percentage of the testing set from total samples	20%
Samples in the testing set	30

The final data set contained 89 failed results and 59 passed samples. The failed class contained samples of broken sensors and beam patterns with the wrong location. The description of the data set is summarised in Table 1.

4.5. Implementation to Find a Suitable Model for Beamforming

At the beginning of the implementation, we collected SS-RSRP measurements, test IDs, beam IDs, and x and y location values from the raw data samples. All SS-RSRP measurements were set in the same order in each beam using location information for x and y, after which the 2D data structure was dismantled into a 1D format. In the same test run, all vectorised beams were concatenated into one long feature vector. The feature vector contained a test ID that made it possible for all preprocessed data to merge their manually collected labels.

After the preprocessing operation, the test ID column was dropped from the data set, and currently, the data set contained 3,906 features. The number of features was large, and therefore, a dimensionality reduction to the data set was performed using principal component analysis. To avoid selecting an arbitrary number of principal components for the learning, we used a 95% cumulative threshold to determine a suitable number of components. However, we did not leave the analysis on it but continued to study the behaviour of the components within the threshold. For the study, we split the data set into training and testing sets at an 80/20 ratio, in which case the training set contained 118 samples, and the testing set contained 30 samples. Before the PCA, the data were standardised using the Scikit-learn library's method *StandardScaler*, which standardises features by removing the mean and scaling to the unit variance. The scaler was built using the training set, and the testing set was scaled into the same form as the training set by using the built scaler.

To select the valid number of PCs, we used repeated 10-fold cross-validation, and the repetition was made 10 times in the cross-validation. In the random forest classifier, we did not focus on finding optimal hyperparameters for the model, and we used the Scikit-learn library's *RandomForestClassifier* method with its default parameters. The default parameters are listed in Appendix 1. After validation, all results were collected to determine the most suitable number of principal components. This thesis presents the results in the subsequent chapter, *Results (Chapter 5.)*.

To confirm the performance of the random forest model, we collected the same results from another method and built the decision tree classifier using the Scikit-learn library's *DecisionTreeClassifier* method with its default parameters. The default parameters of *DecisionTreeClassifier* are presented in Appendix 2. Since the implementation contained many stages, some of which occurred in parallel, Figure 15 offers a straightforward depiction of all the main stages of the implementation with a short description.

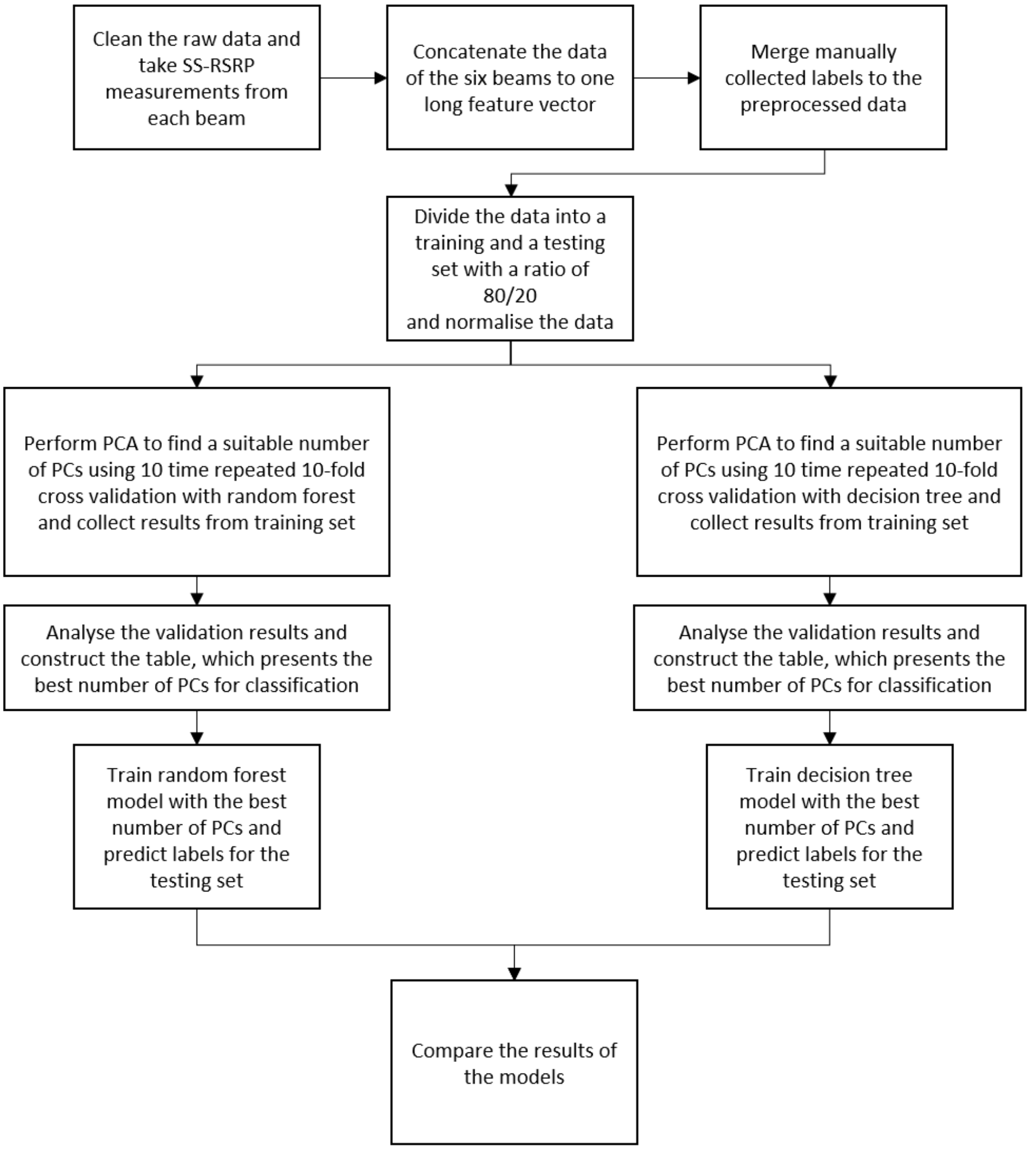

Figure 15. Illustrative figure of the model implementation and its main stages.

5. RESULTS

This chapter presents all the results from the implemented phases, which produced information regarding the performance of the model. Furthermore, this chapter provides the results of the principal component analysis and compares the random forest classifier with the alternative method and another simpler machine learning solution, the decision tree classifier.

5.1. Principal Component Analysis

After the preprocessing, all necessary features were collected from the data, and samples were concatenated to form long feature vectors. However, the feature vector contained a large number of features, and therefore, we decided to use a dimensionality reduction technique to simplify the data set. For the reduction, we used principal component analysis. In the PCA, we used the 95% cumulative variance threshold to avoid arbitrarily choosing the number of principal components. The PCA found 12 principal components to achieve the 95% cumulative variance threshold, and the results are visualised in Figure 16.

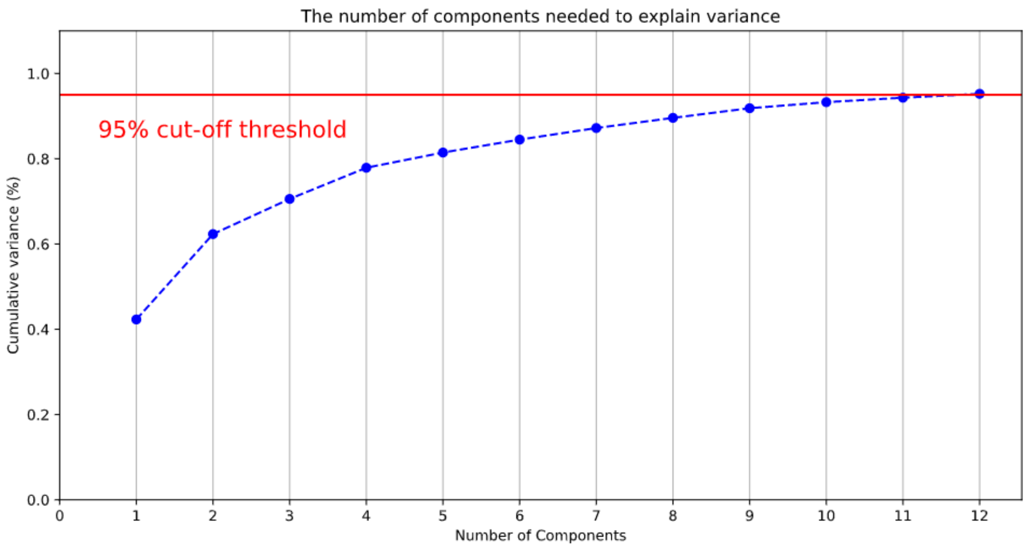

Figure 16. The graph includes all 12 of the most important principal components, their cumulative variance reaching the 95% threshold.

After finding all principal components in the cumulative threshold, we made a comparison between the different numbers of principal components to study how they affect the performance of the model. The number of principal components was selected to confirm and ensure that the number of components would be suitable for the beamforming data and that the model could classify the samples as accurately as possible. The PCA method found a total of 88 principal components. Generally, the number of principal components would be the same as the number of features. Currently, the data set contained 148 samples, of which 88 samples were divided for the training set. In the implementation, we used the Scikit-learn library's PCA method; its documentation explains that the number of components to keep is based on the formulation *n_components == min(n_samples, n_features)* [61]. Therefore, the total number of principal components is 88. To confirm the performance of the random forest classifier, we collected information using all 88 principal components. Moreover, we collected information on the models that utilised 20 and 50 principal components to draw conclusions regarding how the number of principal components affects the random forest classifier and its performance to classify the samples.

To validate the number of principal components, we repeated a 10-fold cross-validation 10 times for the training set, and all results for the different number of principal components are collected in Table 2. For the table, we collected the mean and standard deviation of the accuracy, precision, recall, and F1-score.

Table 2. The table presents the results of the random forest performance, containing the accuracy, precision, recall, and F1-scores for the 10-fold cross-validation repeated 10 times.

The results of the random forest model after repeating the 10-fold cross-validation 10 times using a different number of principal components								
N = PCs	Accuracy		Precision		Recall		F1-score	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
N = 88	0.946	0.059	0.944	0.136	0.912	0.160	0.917	0.126
N = 50	0.959	0.055	0.948	0.129	0.939	0.149	0.936	0.123
N = 20	0.963	0.056	0.946	0.108	0.959	0.100	0.948	0.089
N = 12	0.953	0.060	0.949	0.106	0.937	0.116	0.936	0.089
N = 11	0.954	0.056	0.940	0.117	0.945	0.111	0.935	0.093
N = 10	0.959	0.053	0.942	0.110	0.954	0.101	0.941	0.083
N = 9	0.957	0.056	0.945	0.110	0.946	0.105	0.938	0.085
N = 8	0.957	0.055	0.944	0.113	0.947	0.112	0.938	0.091
N = 7	0.954	0.057	0.947	0.109	0.940	0.112	0.935	0.086
N = 6	0.963	0.052	0.951	0.098	0.966	0.087	0.952	0.072
N = 5	0.965	0.055	0.945	0.114	0.964	0.101	0.948	0.092
N = 4	0.966	0.052	0.954	0.103	0.967	0.101	0.952	0.084
N = 3	0.968	0.050	0.953	0.105	0.969	0.087	0.954	0.078

To understand the content of the table, we now explain the meaning of the metrics. In brief, precision refers to the number of correctly identified samples of a class divided by all samples classified to the same class [62]. Meanwhile, recall refers to the number of samples of the class that the classifier identified correctly divided by the total number of samples in the same class [62]. The final metric, the F1-score, is the combination of these two metrics and serves as a quick means of determining the model's success in its performance [62]. To clarify, the table, precision, recall, and F1-score are average measurements over the passed and failed class representations.

Table 2 reveals that the differences in the performance of the different number of components are relatively small. However, it is observable that generally, all metrics grow when fewer components are used. In the table, N = 3 yielded the greatest average of 0.968 with the lowest standard deviation of 0.050; additionally, its other metrics, such as precision, recall, and especially F1-score are the greatest measurements in Table 2. Therefore, we selected three principal components for the final model and its evaluation.

To better understand the behaviour of the three principal components, we illustrated the components in three figures, with every component plotted against one another. Figure 17 presents the behaviour of the principal components in the training set, in which we can observe clear areas between classes, especially in the two first images, where principal component 1 is plotted against principal components 2 and 3. Still, it can be noticed that some data points are grouped with different classes. This phenomenon can be explained in that the data set consists of samples that contain individual sensor errors. The individual sensor errors occurred when the sensor of the test environment was breaking, thereby causing too large a value from outside the beam pattern. It appears that the classifier ignores individual deviated measurement points in the data.

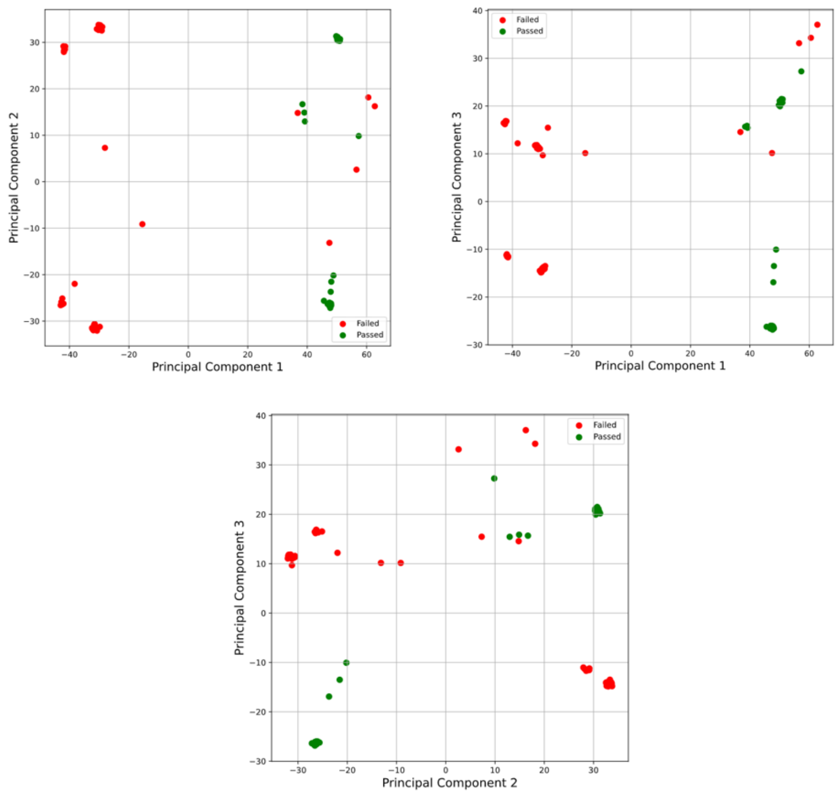

Figure 17. Collection of images, in which three of the most important principal components are plotted against one another. In the images, we can see the categories grouped by classes. Still, some overlap between categories is observable.

5.2. Performance of Random Forest after Dimensionality Reduction

As found in the previous chapter, after validation, the best performance was achieved by using three principal components. For the analysis of the random forest performance, we collected the same metrics for the testing as in the validation. However, this time, we performed a classification report with the testing results, and all data is gathered in Table 3. In addition, we visualised the test results as a confusion matrix, presenting the classification results more effectively. The current data contained two labels, due to which the confusion matrix dimension was 2 x 2. In the classification matrix, the largest values should occur on the diagonal axis, describing how successful the classification is. The confusion matrix of the test results is presented in Figure 18.

Table 3. Classification report of the final random forest model

Random forest classification report				
	Precision	Recall	F1-score	Support
Failed	1.00	0.94	0.97	18
Passed	0.92	1.00	0.96	12
Accuracy			0.97	30
Macro avg	0.96	0.97	0.97	30
Weighted avg	0.97	0.97	0.97	30

The classification of the random forest model reveals that the model performed very well. All collected metrics are balanced, and the F1-score is similar to the accuracy, both of which have a score of 0.97. In addition, the testing set does not contain a large imbalance, and the difference in the number of classes is relatively small. In addition, the confusion matrix reveals that the random forest model predicted one sample incorrectly. To interpret the matrix, the right side indicates the actual classes, where 0 indicates the failed class, and 1 indicates the passed class. The model predicted the passed label instead of the actual failed label for one sample. Overall, the results justify that random forest is a suitable solution for the current data set and can classify quite well.

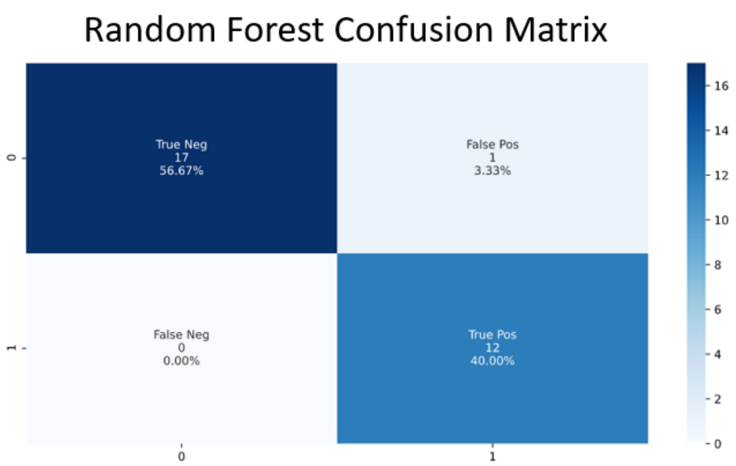

Figure 18. The confusion matrix of the random forest classifier.

5.3. Random Forest Model versus Decision Tree

To confirm that random forest is a good solution for analysing beamforming samples, we tested the use of another, simpler machine learning method: decision tree. For the comparison, we collected the same metrics of the decision tree classifier parallel in the experiment, and all information is gathered in Table 4. The table reveals that the greatest accuracy of the decision tree classifier was achieved using N = 4, which is 0.001 greater than the accuracy achieved by using N = 3. However, with N = 3, the average of the F1 score was higher, and the standard deviation was lower; therefore, we chose three principal components for the decision tree model to compare the performance of the models.

Table 4. The table presents the results of the decision tree performance, containing the accuracy, precision, recall, and F1 scores from the 10-fold cross-validation repeated 10 times.

The results of the decision tree model after repeating the 10-fold cross-validation 10 times using a different number of principal components								
N = PCs	Accuracy		Precision		Recall		F1-score	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
N = 88	0.950	0.066	0.927	0.143	0.941	0.148	0.927	0.132
N = 50	0.958	0.057	0.939	0.142	0.940	0.146	0.933	0.129
N = 20	0.955	0.058	0.933	0.143	0.942	0.142	0.931	0.127
N = 12	0.955	0.058	0.948	0.101	0.940	0.111	0.937	0.083
N = 11	0.959	0.054	0.946	0.109	0.952	0.104	0.941	0.083
N = 10	0.957	0.058	0.949	0.107	0.949	0.100	0.942	0.083
N = 9	0.960	0.056	0.945	0.110	0.955	0.099	0.943	0.083
N = 8	0.954	0.057	0.941	0.116	0.945	0.110	0.936	0.092
N = 7	0.959	0.060	0.946	0.113	0.950	0.113	0.941	0.094
N = 6	0.959	0.056	0.950	0.102	0.953	0.100	0.945	0.079
N = 5	0.965	0.054	0.953	0.099	0.965	0.091	0.954	0.076
N = 4	0.967	0.052	0.948	0.113	0.964	0.097	0.950	0.087
N = 3	0.966	0.056	0.955	0.097	0.966	0.105	0.954	0.086

For the comparison, the training set that contained three principal components was fit to the decision tree model, and the testing set was used to demonstrate the performance. The results of the performance of the decision tree model are gathered in Table 5 and are visualised as a confusion matrix in Figure 19.

In Table 5, the classification report indicates that the accuracy of the decision tree is slightly smaller than in the random forest model, and the confusion matrix in Figure 19 reveals that the difference is very small in practice. The decision tree classifier failed to predict one sample more than the random forest classifier, and the confusion matrix reveals that the decision tree model failed to classify one passed sample as a failed sample, and vice versa.

Table 5. Classification report of the final decision tree model.

Decision tree classification report				
	Precision	Recall	F1-score	Support
Failed	0.94	0.94	0.94	18
Passed	0.92	0.92	0.92	12
Accuracy			0.92	30
Macro avg	0.93	0.93	0.93	30
Weighted avg	0.93	0.93	0.93	30

However, it appears that a simpler algorithm is a suitable solution for classifying the beamforming sample, especially in the current data set. Nevertheless, we must remember that not much data was available, and the data set missed more rare failure cases, as presented in *Three Main Features in Beamforming (Chapter 3.)*.

Overall, the results indicate that random forest performed better in this case. However, it cannot be said that random forest would be better than decision tree, because the differences were too small. Still, the results justify that decision tree is a potential alternative supervised learning algorithm for classifying beamforming samples.

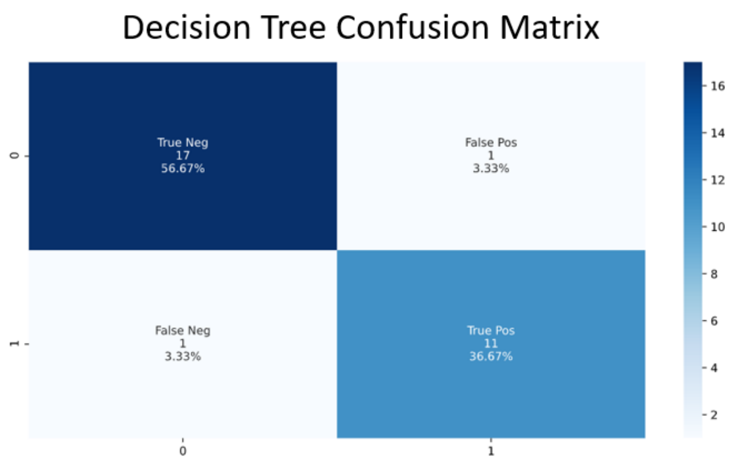

Figure 19. Confusion matrix of the decision tree classifier.

5.4. Machine Learning Approach versus the Alternative Method

When the machine learning approach and alternative method were performed, the machine learning approach was the better of the two. The machine learning approach made individual misclassifications, whereas the alternative method did not provide any total clear green images. However, a small amount of difference between the polarities was present, and therefore, we could interpret the alternative method results and found 14 failed samples and 64 passed samples. In the interpretation, the passed class included a maximum of five individual measurements, where the polarities had a difference of more than 3 dBm.

As we can notice, the alternative method had a different number of samples than the machine learning approach because the methods used different perspectives to analyse the beam patterns. In the machine learning approach, the method is interested only in individual patterns without knowing about the polarities but considering other samples. Instead, the alternative method is interested in only one test sample that contains both polarities. The alternative method focusses on the polarities covering an area equivalently, and therefore, it utilises an entire test run for analysis, whereas the machine learning approach can split data from polarities into two separated samples. Therefore, the results of the methods cannot be compared directly.

However, when comparing these two methods and their respective performance to analyse the samples, the alternative method was too strict. In particular, the alternative method found an individual measurement point in every sample, in which SS-RSRP differed from one another over 3 dBm between the polarities, and therefore, the method was not suitable for analysing beam patterns alone. In addition, the alternative method cannot recognise that the beam pattern is tilting towards the wrong location. The alternative method only takes a stand for polarisation comparison, and it can compute and check the cell size of the beam pattern. Therefore, this implementation is not suitable for automatically classifying the samples into passed or failed classes, but the method provides important additional information regarding the computed beam pattern. Instead, the machine learning approach can locate whether the pattern is in the wrong place, but it does not take a stand if two polarities provide a similar pattern. Overall, the machine learning approach works better for the classification problem, but together, these two methods offer more information about the performance of the beam pattern.

6. DISCUSSION

The results indicated that the machine learning approach can be utilised to analyse beamforming, particularly data from GoB. The random forest algorithm is one of the most popular classification methods used today, and the results indicate that the algorithm works quite well with the collected data. Although the research yielded great results regarding the performance of the algorithm, it must be noted that there was not much data available for the analysis. However, the study justifies that it is promising to employ machine learning methods to analyse beamforming and research how to integrate the developed model into the test environment.

Furthermore, the results prove that principal component analysis found similar patterns between the samples to reduce the original features into a few principal components. It proves that the data set can be highly simplified for training and that the data set does not require a high compute capacity. The features could be reduced from 3,906 to only three principal components in the preprocessed data set, a significant change. The PCA helped to review the data set and its behaviour. The figures reveal the behaviour of the data, which can be classified using a few principal components from the original data. This is because the original features correlate strongly in the original data set, and therefore, it is possible to utilise only three principal components for the classification.

In analysing the performance of the random forest algorithm in the beamforming samples based on the results, they justify that the method can perform classification very well. The accuracy and other metrics are very good, although the data set had a minor imbalance between the occurrences of the classes. The reason why individual samples were misclassified is that the variability in the values of successful samples is small, and a classifier can primarily notice significant changes, such as an abnormal location, in a sample. However, a small deviation, such as a broken sensor, might be too difficult to notice in the statistics when it, for example, affects only one measurement point, thereby appearing unmarked for classification. In addition, it is possible that the data set contained samples, thus causing some misclassifications. Nevertheless, these misclassifications did not dramatically affect the performance of the random forest algorithm.

Although the results reveal that the random forest algorithm is a suitable solution for the current data set, it must be noted that the data did not contain a large number of samples. In addition, the diversity of the error cases is relatively small, and the data contained only cases regarding broken sensors and unexpected locations, though this thesis explains a variety of error cases beyond those experienced here. Therefore, further research is needed to provide an improved model for the beamforming analysis, which can detect and name the different error scenarios. However, the results indicate that random forest, or even a simpler classifier such as decision tree, can be a suitable solution for analysing beamforming samples, which provides an excellent premise to continue the research.

Furthermore, this work justifies that the machine learning method provides more relevant feedback than the introduced alternative method. Still, it must be noted that these two methods have different perspectives in analysing beamforming samples. However, together, the methods can offer complementary information regarding beamforming on the radio, which can be more beneficial for an end-user.

6.1. Further Studies for Beamforming Analysis

The original idea was to use throughput as a feature for training, but unfortunately, this information was not available for the research. Therefore, it needed to be omitted in the implementation, which focussed on analysing the pattern of the beam set by utilising the machine learning approach. However, throughput would provide more detailed information regarding how well a radio can serve user equipment. The current implementation takes a stand only in that the beam pattern is formulated correctly.

The solution of the project concentrated only on analysing one radio product and one pattern of beams. The other remarkable approach for future research is to study a general analysis for radio variants and beam patterns so that there is no need to train dozens of models for each case. For example, it can be wondered whether it is possible to assume that every radio variant and beamforming type offers a similar pattern and builds a more generalised analysis model.

In addition, research should be conducted on how to implement automated learning processes directly in automated testing environments, a topic into which this thesis does not investigate. The final model was integrated into a separate application to analyse the newly emerging samples from the same radio type. The application can be updated by adding the new models to it, but this phase occurs manually. Therefore, there is a need to investigate a suitable automated solution for the entire automatic analysis process.

6.2. Alternative Solutions

This thesis focusses on taking the first step towards utilising machine learning in beam pattern analysis. Therefore, we concentrated on using the simplest machine learning methods to analyse the data. For the implementation, not a large amount of data was available. Still, the approach was suitable for the current situation and proved that machine learning is a valuable tool for automated analysis.

Nevertheless, consideration should be taken to use more advanced techniques to analyse beamforming, such as neural networks and computer vision-type methods, in the future. These approaches would be more powerful in solving the problem, but the currently available data restricts the use of these methods. If more data is available for future research or there is a convenient way to simulate data, deep learning and computer vision techniques should be tested and compared with the results of this thesis.

Although we were unable to use the previously mentioned approaches for the current data set, it would be necessary to improve the solutions and compare other machine learning methods and how they fit the analysis or how the current implementation can be improved, for example, to concentrate on tuning hyperparameters. Different methods could be tested, and their performance with respect to one another could be compared. As in this thesis, the results of the decision tree algorithm reveal that it could be used as a potential alternative algorithm for classifying beamforming samples.

The other suggestion for using basic machine learning methods is to approach the problem from a different perspective. In this thesis, we used only one classifier to make classifications. However, it could be done such that all beams are not concatenated into one long feature vector, and every separated beam could have its own classifier. Then, they could be used for decision-level classification between beams to analyse the total behaviour of the beam pattern.

As we can see in this chapter, the thesis work is merely the beginning of beamforming analysis, and it is possible to continue and expand the work through several different approaches.

7. CONCLUSION

The research question of this thesis was to study whether it is possible to analyse beamforming using the existing machine learning method. Beamforming consists of many complicated mathematical calculations to compute beam patterns and a variety of architectures to perform beam patterns. Therefore, this thesis presents an idea to simplify beamforming into three main features—the polarities, the pattern of beam set, and the throughput—which can be utilised for analysis. This idea makes it possible to utilise features, which always appear regardless of the type of beamforming.

Unfortunately, throughput measurements were not available for this thesis, and therefore, we needed to concentrate on analysing only the pattern of the beam set. This thesis introduced an alternative method, which likewise utilised information about the determined high-level features. The method utilised information regarding the difference in polarities and the size of the beam. Nonetheless, the alternative method was not sufficient to provide information to make independent classifications for the samples. Therefore, a machine learning approach was selected for this thesis, and we determined that the approach could complement the beamforming analysis and that machine learning could be utilised to make independent classifications in a test environment.

In this thesis, the data was collected from the radio, which utilised the GoB technique to compute a stable beam pattern. We selected the random forest algorithm to determine whether the machine learning approach is a more suitable solution than the alternative method for classifying beamforming samples. Although not many samples were available and the data set contained a small imbalance between the classes, the results proved that random forest could classify beamforming results very well without overfitting occurring in the results. In addition, the research revealed that beamforming classification does not depend on the random forest method, and it is possible to utilise other classification methods, such as decision tree, which is a more straightforward method than random forest.

As a final word, this thesis provides various directions for improving the results and finding a more suitable solution. In addition, the results of this thesis serve as motivation to achieve a more accurate model providing comprehensive and detailed feedback for the end-user. The demand for reliable communication increases at all times, and it will grow increasingly more complex. Therefore, the automated analysis and its feedback are needed to provide a more efficient environment in radio development.

8. REFERENCES

[1] Hossain S. (2013) 5g wireless communication systems. American Journal of Engineering Research 2, pp. 344–353.

[2] Singh H. & Singh B.P. (2013) Business intelligence: Effective machine learning for business administration. International Journal of IT, Engineering and Applied Sciences Research 2, pp. 13–19.

[3] Parodi F. (1989) 19 - physics and chemistry of microwave processing. In: G. Allen & J.C. Bevington (eds.) Comprehensive Polymer Science and Supplements, Pergamon, Amsterdam, pp. 669–728.

[4] Lorenz M. (2018), How 4x4 mimo can boost your mobile network. URL: https://www.rohde-schwarz.com/us/solutions/test-and-measurement/mobile-network-testing/stories-insights/article-how-4-4-mimo-can-boost-your-mobile-network_253229.html. Accessed 3.4.2021.

[5] Bellis M. (2020), The history of radio technology. URL: https://www.thoughtco.com/invention-of-radio-1992382. Accessed 3.4.2021.

[6] Van Veen B. & Buckley K. (1988) Beamforming: A versatile approach to spatial filtering. IEEE ASSP Magazine 5, pp. 4–24.

[7] Haynes T. (1998) A primer on digital beamforming. Spectrum Signal Processing , pp. 1–15.

[8] Fruhlinger J. (2019), Beamforming explained: How it makes wireless communication faster. URL: https://www.networkworld.com/article/3445039/beamforming-explained-how-it-makes-wireless-communication-faster.html. Accessed 3.4.2021.

[9] Ali E., Ismail M., Nordin R. & Abdulah N.F. (2017) Beamforming techniques for massive mimo systems in 5g: overview, classification, and trends for future research. Frontiers of Information Technology & Electronic Engineering 18, pp. 753–772.

[10] Steyskal H. (1988) Digital beamforming. In: 1988 18th European Microwave Conference, pp. 49–57.

[11] Reil M. & Lloyd G. (2016) Millimeter-wave beamforming: Antenna array design choices and characterization. Tech. rep., Rohde and Schwarz.

[12] Holma H., Toskala A., Nakamura T. & Uitto T. (2019) Introduction, John Wiley and Sons, Ltd, chap. 1. pp. 1–11.

[13] Kim Y.J., Sultan Q. & Cho Y.S. (2020) Pilot sequence design for mmwave cellular systems with relay stations in the presence of blockage. IEEE Access 8, pp. 80454–80467.

[14] GSMA (2020) 5g spectrum gsma public policy position. Tech. rep., GSMA.

[15] Salh A., Audah L., Mohd Shah N.S. & Hamzah s.a. (2020) Energy-efficient power allocation with hybrid beamforming for millimetre-wave 5g massive mimo system. Wireless Personal Communications .

[16] Wang C., Haider F., Gao X., You X., Yang Y., Yuan D., Aggoune H.M., Haas H., Fletcher S. & Hepsaydir E. (2014) Cellular architecture and key technologies for 5g wireless communication networks. IEEE Communications Magazine 52, pp. 122–130.

[17] Passoja M. (2018), 5g nr: Massive mimo and beamforming – what does it mean and how can i measure it in the field? URL: https://www.rcrwireless.com/20180912/5g/5g-nr-massive-mimo-and-beamforming-what-does-it-mean-and-how-can-i-measure-it-in-the-field. Accessed 3.4.2021.

[18] Massive mimo. URL: https://www.nokia.com/networks/solutions/massive-mimo/#overview. Accessed 3.4.2021.

[19] Taruna S.and Kaur I. (2013) Analysis of multiple-input-multiple-output (mimo) system with transmit and receive diversity. International Journal of Computer Applications 79, pp. 24–27.

[20] Powell C. (2014) Technical analysis: Beamforming vs. mimo antennas. Tech. rep., RFS.

[21] Massive mimo and beamforming – what is it? URL: https://future-markets-magazine.com/en/markets-technology-en/what-is-mimo-beamforming/. Accessed 3.4.2021.

[22] Molisch A.F., Ratnam V.V., Han S., Li Z., Nguyen S.L.H., Li L. & Haneda K. (2017) Hybrid beamforming for massive mimo: A survey. IEEE Communications Magazine 55, pp. 134–141.

[23] Ganesan R.S., Zirwas W., Panzner B., Pedersen K.I. & Valkealahti K. (2016) Integrating 3d channel model and grid of beams for 5g mmimo system level simulations. In: 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1–6.

[24] Loy J. (2019) Neural Network Projects with Python. Packt Publishing.

[25] Awasthi T.R., The complete roadmap for ai, data scientist aspirants. URL: https://tekrajawasthi15.medium.com/the-complete-roadmap-to-be-a-data-scientist-9a07721b88fd. Accessed 3.4.2021.

[26] Géron A. (2019) Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media.

[27] Zhu X. (2005) Semi-supervised learning literature survey. Computer Science Department, University of Wisconsin-Madison .

[28] Ghahramani Z. (2004) Unsupervised learning. In: Advanced Lectures on Machine Learning, Springer-Verlag, pp. 72–112.

[29] Dietterich T. (1995) Overfitting and undercomputing in machine learning. ACM Comput. Surv. 27, p. 326–327.

[30] Anagnostopoulos T., Anagnostopoulos C., Hadjiefthymiades S., Kyriakakos M. & Kalousis A. (2009) Predicting the location of mobile users: A machine learning approach. In: Proceedings of the 2009 International Conference on Pervasive Services, ICPS '09, Association for Computing Machinery, New York, NY, USA, p. 65–72.

[31] Weisberg S. (2005) Applied Linear Regression. Wiley, Hoboken NJ, third ed.

[32] Kotsiantis S. (2007) Supervised machine learning: A review of classification techniques. Informatica (Slovenia) 31, pp. 249–268.

[33] Breiman L. (2001) Random forests. Machine Learning 45, pp. 5–32.

[34] Lan H. & Pan Y. (2019) A crowdsourcing quality prediction model based on random forests. In: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), pp. 315–319.

[35] Rokach L. & Maimon O. (2005) Decision Trees, Springer US, Boston, MA, chap. 9. pp. 165–192.

[36] Bhargava N., Sharma G., Bhargava R. & Mathuria M. (2013) Decision tree analysis on j48 algorithm for data mining. International Journal of Advanced Research in Computer Science and Software Engineering .

[37] Murty M. & Devi V. (2011) Pattern Recognition. Springer.

[38] Bramer M. (2007) Avoiding Overfitting of Decision Trees, Springer London, London, chap. 8. pp. 119–134.

[39] Liberman N. (2017), Decision trees and random forests. URL: `https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991`. Accessed 3.4.2021.

[40] Raj J.T. (2019), A beginner's guide to dimensionality reduction in machine learning. URL: `https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e`. Accessed 3.4.2021.

[41] Abdi H. & Williams L.J. (2010) Principal component analysis. WIREs Comput. Stat. 2, p. 433–459.

[42] Piippo P. (2018) Automatic Capacity Test Case Result Analysis From CPU Loads. Master's thesis, University of Oulu.

[43] Iso-Junno I. (2018) Massadata-analyysijärjestelmä Suoritin- Ja Muistinkäytölle. Master's thesis, University of Oulu.

[44] Stoughton J. (2018), How machine learning is creating new opportunities in the telecommunications industry. URL: https://www.digitalistmag.com/digital-economy/2018/07/18/how-machine-learning-is-creating-new-opportunities-in-telecommunications-industry-06179697/. Accessed 3.4.2021.

[45] Geotel, 5 ways artificial intelligence and machine learning are used in telecom. URL: https://www.geo-tel.com/5-ways-artificial-intelligence-and-machine-learning-used-telecom/. Accessed 3.4.2021.

[46] Fournier A.F. (2019) Recommendation on Base Station Antenna Standards (V11.1). Tech. rep., NGMN.

[47] Gao X., Cui Y. & Li R. (2016) Broadband vertically/horizontally dual-polarized antenna for base stations. In: 2016 11th International Symposium on Antennas, Propagation and EM Theory (ISAPE), pp. 79–80.

[48] 3GPP, 5g; nr; requirements for support of radio resource management (3gpp ts 38.133 version 15.3.0 release 15). URL: https://www.etsi.org/deliver/etsi_ts/138100_138199/138133/15.03.00_60/ts_138133v150300p.pdf. Accessed 3.4.2021.

[49] Moore B.C. (1995) Hearing. Academic Press, second ed.

[50] DNSstuff (2019), What Is Throughput in Networking? Bandwidth Explained. URL: https://www.dnsstuff.com/network-throughput-bandwidth. Accessed 3.4.2021.

[51] Andrews J.G., Ganti R.K., Haenggi M., Jindal N. & Weber S. (2010) A primer on spatial modeling and analysis in wireless networks. IEEE Communications Magazine 48, pp. 156–163.

[52] Li H. & Ye W. (2015) Study of digital modulation signal error vector magnitude based on vector signal analyzer. In: 2015 8th International Symposium on Computational Intelligence and Design (ISCID), vol. 2, vol. 2, pp. 100–103.

[53] Nguyen V.M. & Baccelli F. (2010) A stochastic geometry model for the best signal quality in a wireless network. In: 8th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, pp. 465–471.

[54] Python. URL: https://www.python.org/. Accessed 3.4.2021.

[55] McKinney W. (2018) Python for Data Analysis. O'Reilly, second ed.

[56] pypi.org, pandas 1.0.5. URL: https://pypi.org/project/pandas/. Accessed 3.4.2021.

[57] scikit learn.org, scikit-learn machine learning in python. URL: https://scikit-learn.org/stable/. Accessed 3.4.2021.

[58] Jain K. (2015), Scikit-learn(sklearn) in python – the most important machine learning tool i learnt last year! URL: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/. Accessed 3.4.2021.

[59] matplotlib.org, Matplotlib: Visualization with python. URL: https://matplotlib.org/. Accessed 3.4.2021.

[60] seaborn.org, seaborn: statistical data visualization. URL: https://seaborn.pydata.org/. Accessed 3.4.20211.

[61] scikit learn.org, sklearn.decomposition.pca. URL: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html?highlight=pca#sklearn.decomposition.PCA. Accessed 3.4.2021.

[62] Kreiger J. (2020), Evaluating a random forest model. URL: https://medium.com/analytics-vidhya/evaluating-a-random-forest-model-9d165595ad56. Accessed 3.4.2021.

9. APPENDICES

In the thesis, the random forest classifier used Scikit-learn library's *RandomForestClassifier* method with following default parameters

- n_estimators=100,
- criterion='gini'
- max_depth=None
- min_samples_split=2
- min_samples_leaf=1
- min_weight_fraction_leaf=0.0
- max_features='auto'
- max_leaf_nodes=None
- min_impurity_decrease=0.0
- min_impurity_split=None
- bootstrap=True
- oob_score=False
- random_state=None
- verbose=0
- warm_start=False
- class_weight=None,
- ccp_alpha=0.0
- max_samples=None

In the thesis, the decision tree classifier used Scikit-learn library's *DecisionTreeClassifier* method with following default parameters

- criterion='gini'
- splitter='best'
- max_depth=None
- min_samples_split=2
- min_samples_leaf=1
- min_weight_fraction_leaf=0.0
- max_features=None
- random_state=None
- max_leaf_nodes=None
- min_impurity_decrease=0.0
- min_impurity_split=None
- class_weight=None
- ccp_alpha=0.0