Reducing Inter-cell Interference Using Machine Learning

ALBIN FRISK
MASTER´S THESIS
DEPARTMENT OF ELECTRICAL AND INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING | LTH | LUND UNIVERSITY

Reducing Inter-cell Interference Using Machine Learning

Albin Frisk
al2217fr-s@student.lu.se

Department of Electrical and Information Technology
Lund University

Supervisors: Ove Edfors, Lund University
Sarthak Das, Ericsson AB
Veluru Chaithanya Kumar Reddy, Ericsson AB

Examiner: Fredrik Rusek

September 18, 2021

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

One way of meeting the increasing demand for higher data rates is by building denser cellular networks in order to maximize the use of the frequency spectrum. The denser deployment leads to an increased probability of inter-cell interference, the phenomenon where the signal quality experienced by a user served by a cell is lowered because of the transmission from neighboring cells in the network. Coordinated multi-point techniques such as dynamic point blanking (DPB) can be used to dynamically mute resources in the network and increase the possible capacity in the system. The possible permutations of muting patterns are in this thesis found by using a search tree structure and the optimal pattern is found by evaluating each node in the tree.

In some scenarios it not necessary to evaluate all nodes in order to find a muting pattern that results in satisfying performance, making it possible to save computational power. In this thesis the three machine learning models, logistic regression, support vector machine and naive Bayes, have been used to binary classify the search width needed in order to decrease the mean interference experienced by users below a certain threshold. The data needed to train the models was generated in an Ericsson simulator, where features were extracted from the output logs.

Results show that the support vector machine is the most successful of the three and it was able to predict the search width in 87 % of the samples. The conclusion is that machine learning techniques can be used to predict the optimal search depth in a given scenario, but more research has to be done in order to quantify the computational gain in reducing the number of evaluated muting patterns.

Populärvetenskaplig sammanfattning

I det här arbetet har det undersökts hur maskininlärningstekniker kan användas för att optimera tekniker som används för att minska inter-cell interferens. Inter-cell interferens är ett fenomen som uppstår i de mobila nätverken eftersom basstationerna är uppdelade i olika celler där alla celler använder samma resurser när de sänder data till mobilanvändarna. Det kan då uppstå störningar hos mobilanvändarna på grund av att de plockar upp signaler från närliggande celler som var menade att plockas upp av en annan användare.

Det finns ett flertal tekniker för att minska mängden inter-cell interferens i de mobila nätverken. Tekniken som används i den här studien kallas "Dynamic point blanking" och innebär att man helt enkelt slutar använda vissa resurser som anses bidra till mycket interferens. Utmaningen i den här lösningen är att veta vilka som är de resurser som är är mest optimala att inte använda vid en viss tidpunkt och ett sätt att hantera detta är att utvärdera flera olika kombinationer av hur resurserna används.

Det är kostsamt att utvärdera dessa kombinationer och det är därför av intresse att minska hur många alternativ som ska utvärderas vid scenarion då vinsten av att utvärdera alla kombinationer inte är särskilt stor. I denna studie utforskades hur maskininlärning kan användas för att identifiera vid vilka tillfällen då det är värt att utvärdera färre eller flera kombinationer.

Maskininlärningsmodeller lär sig hitta mönster utifrån data. Datan är strukturerad på ett sådant sätt så att varje datapunkt innehåller ett antal parametrar som är kopplade till ett rätt svar. Genom att gå igenom en stor mängd data så kan maskinen hitta ett mönster och på så sätt förutse hur många kombinationer som är värda att utvärdera vid en viss tidpunkt.

I det här fallet så genererades datan i ett simuleringsprogram och i simuleringarna användes ett mobilt nätverk uppbyggt av tre celler. Maskinen fick basera sina gissningar på antalet användare som befinner sig i nätverket, hur mycket resurser som används i de tre cellerna och vinkeln på antennerna i cellerna. Det slutgiltiga resultatet blev att maskinen kunde gissa rätt i cirka 87 % av fallen och kan således användas för att effektivisera de mobila nätverken.

Acknowledgements

I would like to start by thanking Bhavin Patel, my manager at Ericsson, for giving me this opportunity of writing this thesis and giving me all the resources needed in order to finish it.

I want to thank Chaithanya and Sarthak, my supervisors at Ericsson, for always being there for me when I needed help, and providing great feedback at all stages during the project.

I would like to thank Ove from Lund University for believing in my abilities to finish this thesis and for going out of his way to make it possible to actually finish it in time.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

This chapter starts by providing a background and motivation to the work done in the thesis. The purpose is given followed by an overview of the method used to generate the results. The last section provides the disposition of the thesis.

1.1 Background and Motivation

The ability to reuse frequencies is one of the core features of cellular networks, but this feature also leads to interference between cells. Early cellular networks used a static frequency separation where neighboring cells transmitted on different frequency bands in order to mitigate this interference. This may have been a satisfying solution at the time, but since then the demand for higher data rates has increased substantially. The demand for higher data rates in the mobile networks therefore push operators to abandon this static frequency separation technique, leading to increased inter-cell interference. The need for more efficient use of the spectrum also drives denser cell deployment, further increasing the risk of interference.

Coordination between cells is a technique that can be used to deal with the inter-cell interference. Coordination can be implemented in such a way that the coordinating cells agree on which cell should and should not transmit on some part of the spectrum at each time interval. This way of coordinating is known as dynamic point blanking. The challenge with dynamic point blanking is how to decide on which muting pattern to choose. One way to handle this is to evaluate a number of muting patterns and see which one will result in the best outcome. In this thesis it will be studied how machine learning can be used to intelligently decide on the amount of muting patterns to evaluate.

1.2 Purpose

The purpose of the thesis is to use machine learning algorithms to identify scenarios where it is worthwhile or not worthwhile to evaluate more or less muting patterns. The aim is that this will result in more efficient use of the available resources and therefore lead to better overall network performance.

1.3 Method

The data needed to train the machine learning models was generated in an Ericsson simulator. The machine learning approach for solving the problem was to use supervised learning models and the data therefore had to be labeled and a set of features had to be found. The features used in the thesis were found by studying simulation logs and identifying which parameters that had an impact on the label of a data sample. Three models were trained and evaluated using the data samples from the generated dataset in order to be able to find a suitable model for the given problem.

1.4 Disposition

Chapter 2 goes through the telecommunication theory relevant for understanding the telecommunication aspects of the thesis. Chapter 3 gives a theoretical background to machine learning and explains the models and validation techniques used. Chapter 4 presents the method that was used to gather the results. Chapter 5 presents the achieved results when using the different machine learning models. Chapter 6 provides a discussion of the results and propose ideas for what could be interesting to research further.

Chapter 2
Telecommunication Theory

This chapter contains the telecommunication theory that is needed in order to understand the problem presented in the thesis. The first sections provides some basics of OFDM and the LTE time-frequency structure. The SINR and the concept of inter-cell interference (ICI) is described in greater detail. The final section explains dynamic point blanking (DPB), which is the technique that machine learning was applied upon in order to reduce ICI.

2.1 OFDM

Orthogonal Frequency Division Multiplexing (OFDM) is a transmission scheme that is very well suited for transmission of high data rates in time dispersive environments [1]. In OFDM one high data rate stream is divided into N parallel streams which then are transmitted by modulating a set of N carriers, referred to as subcarriers. This increases the symbol time on each subcarrier by a factor N. The signals carried by the different subcarriers has to be orthogonal in order for the receiver to be able to separate them. Unfortunately, time dispersion leads to loss of orthogonality between subcarriers, which in turn leads to inter carrier interference. In order to remove these negative effects, something called the cyclic prefix (CP) is added to the start of each OFDM symbol. The CP is constructed by copying the last part of the OFDM symbol and adding it to the beginning of the symbol. The combination of a long symbol time and a CP makes OFDM very robust against channel-frequency selectivity. It is because of this that OFDM is the modulation scheme that is used in both LTE and NR [2].

2.2 LTE Time-Frequency Structure

As mentioned in the previous section, OFDM is the transmission technique used in the LTE standard. In LTE, the OFDM subcarrier spacing is 15 kHz in both downlink and uplink. This subcarrier spacing was chosen in order to provide a balance between the added overhead of the CP and sensitivity to various types of frequency inaccuracies, Doppler spread and errors.

LTE transmissions are separated into frames of length 10 ms in the time domain, as seen in Figure 2.1. Each frame is separated into ten equally sized subframes of length 1 ms and each subframe is separated into two equally sized slots.

Finally, each slot contains a number of OFDM symbols and their corresponding CPs. LTE specifies two different lengths of the CP, known as normal CP and extended CP. Using normal CP length results in seven OFDM symbols per slot, while extended CP results in six OFDM symbols per slot. In LTE, the smallest

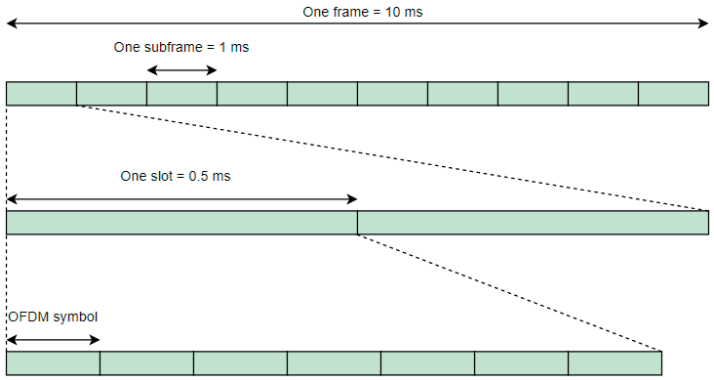

Figure 2.1: LTE time-domain structure when using normal CP length

resource in the frequency domain is the subcarrier of an OFDM symbol. The smallest available physical resource is called a resource element and is equal to one subcarrier during the time of one OFDM symbol. As seen in Figure 2.2, twelve subcarriers in each slot are grouped together into what is referred to as a physical resource block (PRB). A PRB therefore contains $7 \cdot 12 = 84$ resource elements if normal CP is used and $6 \cdot 12 = 72$ resource elements if extended CP is used. In LTE the smallest time-domain unit for dynamic scheduling is one subframe. Therefore the smallest unit available for scheduling is two consecutive PRBs in one subframe, sometimes called a resource-block pair.

2.3 SINR

Signal-to-interference-and-noise ratio (SINR) is a quantity that is used to measure the quality of a received signal or signals and can be used to calculate the upper bound of the possible data rate. The SINR is defined as in (2.1), where P is the received power, I is the power of the received interference signals and N is the noise in the receiver.

$$SINR = \frac{P}{I + N} \tag{2.1}$$

In cellular networks the signal-to-noise ratio (SNR) is often insufficient to model the system because of the non-negligible interference that is present in the network. One cause of the interference is that neighboring cells are transmitting on the same time-frequency resource as the serving cell. The increasing demand for higher data rates drives denser cell deployment, which in turn increases the interference even more. Coordination between cells is used in order to reduce the amount

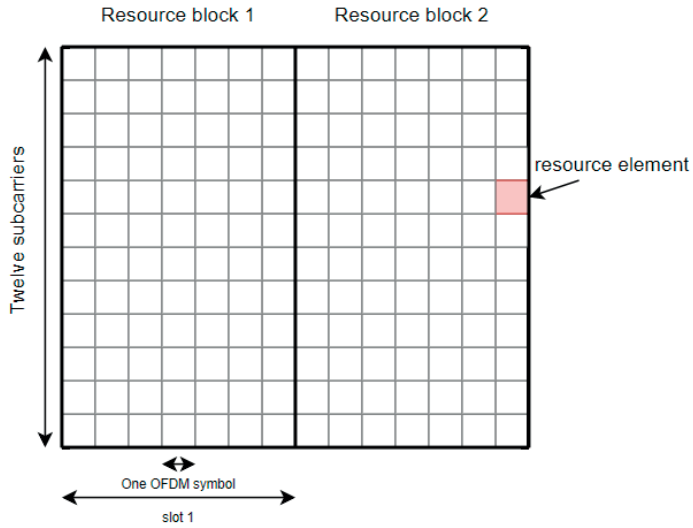

Figure 2.2: LTE time-frequency structure when using normal CP length

of interference measured by the UEs in the network, and therefore increases the network throughput.

2.4 Inter-Cell Interference

Downlink ICI can occur when two neighboring cells are transmitting on the same time-frequency resources. Figure 2.3 shows a scenario where two UEs that are served by two different cells are receiving data on the same time-frequency resource. The rightmost UE is positioned close to its serving cells edge and are therefore receives some of the signal power that are meant for the leftmost UE. It is this signal power from the leftmost cell that are referred to as ICI in this scenario. The interference experienced by the rightmost UE will lower its SINR and therefore lower the upper bound on the achievable data rate for that UE.

As stated in the previous section, coordination between cells can be used to reduce the ICI. This can be done by, for example, avoiding transmitting on a time-frequency resource (known as dynamic point blanking), dynamically adjusting the transmit power (known as coordinated power control) or adjusting the direction of the transmission (known as coordinated beam-forming). In this thesis only techniques for avoiding transmission on the same time-frequencies has been used and will be further explained in the following section.

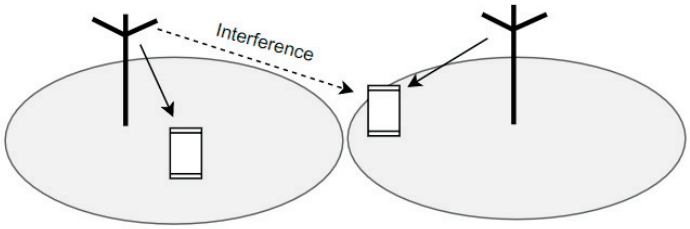

Figure 2.3: Two cells that are transmitting on the same time-frequency resource, causing ICI at the UE

2.5 Dynamic Point Blanking

As previously mentioned, DPB is a technique that can be used in order to reduce the ICI. The main idea behind DPB is to coordinate the transmission decisions between cells. By sharing information and coordination between cells it is possible to identify if two cells are about to use same time-frequency resource and therefore causing ICI for some UE. If a UE is believed to experience significant ICI it is possible to mute the time-frequency resources in the interfering cell, resulting in a higher SINR experienced by the UE. Figure 2.4 shows a scenario where two neighboring coordinating cells have coordinated their time-frequency resources in order to mitigate the ICI that otherwise would be experienced by the two served UEs. In order to effectively perform DPB the network has to be able to estimate

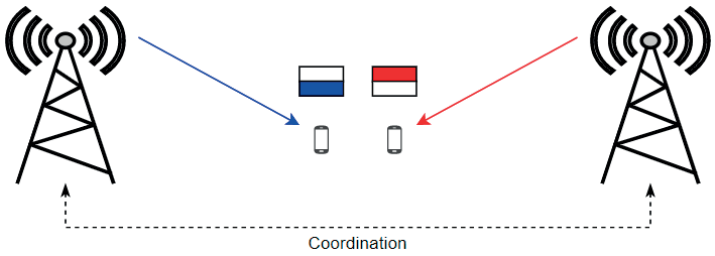

Figure 2.4: DPB illustration

the impact a neighboring cell transmission has on the channel quality of the impacted UE, and how much the channel quality would improve if the transmission in the neighboring cell was to be muted. Transmission mode 10, introduced in LTE release 11, can be used for interference estimation. Transmission mode 10 uses a channel state information interference measurement (CSI-IM), where a CSI-IM configuration specifies the set of resource elements in a subframe that should be used to estimate the interference. By using several CSI-IM configurations it is possible to estimate how much different configurations of transmitting cells impacts the interference experienced by a UE. After the interference has been estimated it

is straightforward to estimate the SINR gain that would be the outcome of muting the resources in the neighboring cells.

In scenarios where there are several UEs it is necessary to take the interference estimations of all UEs into account when scheduling the available resources in order to not make a decision that decreases the signal quality for some UEs. One way to do this is by constructing a search tree where each node is associated with a certain muting decision. The best muting decision can then be found by searching through all the nodes and selecting the muting decision that corresponds to the largest gain.

In some scenarios it is not necessary to search through all nodes in the search tree in order to get a satisfying result. One way to limit the amount of nodes that is evaluated is to use a maximum search width. The search width specifies the amount of nodes that are evaluated at each level in the tree and therefore reduces the number of computations needed in order to decide on a muting decision. Figure 2.5 and Figure 2.6 illustrates how the selected muting pattern changes when the search width is increased from two to three. Muting pattern h14 is the selected muting pattern when search width two is used. When the maximum search width is increased to three the decision is instead h13, a more optimal muting pattern compared to h14.

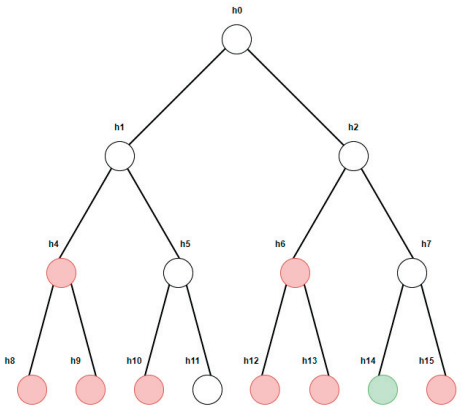

Figure 2.5: Search tree structure using search width two.

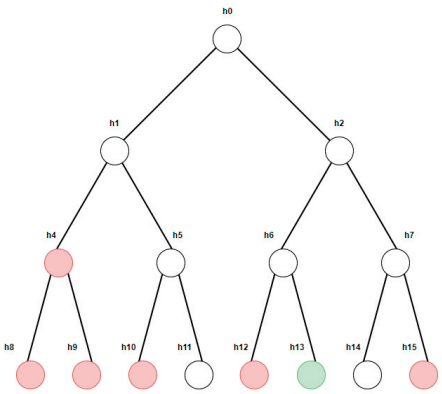

Figure 2.6: Search tree structure using search width three.

Chapter 3

Machine Learning Approach

This chapter goes through all the theory needed to understand the machine learning aspects of the thesis. It starts with an introduction to the area of machine learning, followed by an explanation of the three common approaches: Supervised learning, unsupervised learning and reinforced learning. The theory of the three different models used in the thesis is provided.

3.1 Introduction to Machine Learning

Machine learning (ML) can be regarded as a subfield of artificial intelligence. A ML algorithm is a type of algorithm that can improve by learning from experience and solve problems without being explicitly programmed. ML is useful in situations where it is difficult or impossible to develop algorithms specified for solving the given problem [4].

3.2 Supervised Learning

In supervised learning the machine is given a dataset of data samples together with the correct answer to each data sample as seen in Figure 3.1. The machine will then receive a data sample as input and try to predict what the correct answer is. The guess is then compared to what the provided correct answer was. Depending on if the guess was correct or incorrect, the machine will have to adjust some function parameters in order to increase the probability of making correct guesses in the future. The end goal is that the machine should be able to successfully predict the outcome/right answer of new, unlabeled, data samples.

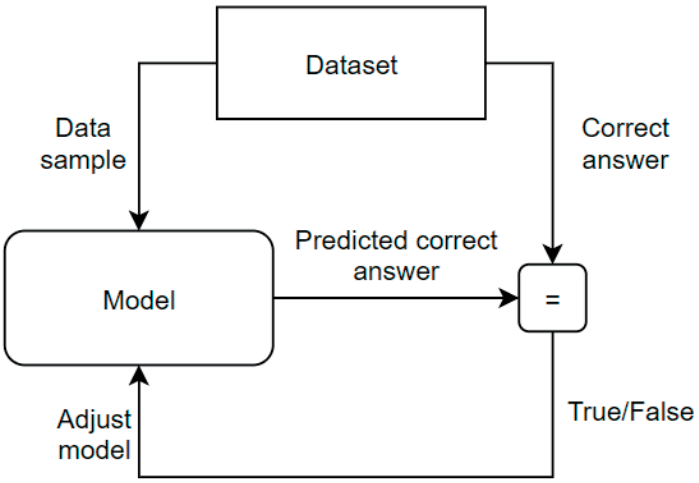

Figure 3.1: An illustration of supervised learning

Supervised learning can be divided into two categories: Classification and regression. Classification refers to the ability to correctly classify something into a set of categories or classes. In regression the goal is to predict real values, for example, the number of customers visiting a store in day.

3.3 Unsupervised Learning

In unsupervised learning the provided data samples in the dataset are not accompanied by correct answers. From the data the machine may find patterns of similarity. The algorithm can then find clusters of similar items in the dataset. The main difference from supervised learning is that unsupervised learning solves the problem of finding a pattern in the dataset, while supervised learning tries to make predictions from new samples that was not included in the provided dataset.

3.4 Reinforcement Learning

Reinforced learning is similar to unsupervised learning in the way that neither relies on provided examples of correct behavior. The main difference is that unsupervised learning tries to find a hidden structure in the provided data, while reinforced learning tries to maximize a reward. In reinforcement learning the machine interacts with the environment and studies which actions provide the highest reward.

3.5 Model Selection

Over the years several models have been developed for the purpose of solving ML problems. There is no single model that reaches the best performance in all cases.

The performance of a model is very dependent on the structure of the underlying data and it is often a difficult task to identify the optimal model by studying the provided dataset. It is therefore a good practice to compare the performance of several models and select the one that is the best fit for the data. In this thesis the three models logistic regression, support vector machine and naive Bayes are compared.

3.6 Logistic Regression

Logistic regression is a linear method for classification and is commonly used in binary classification applications. Logistic regression emerges from the wish to model the posterior probabilities of the two classes by using a linear function of the features $\boldsymbol{x} = x_1, ..., x_n$. This gives the model the form of

$$\log \frac{p(C = +|x_1, .., x_n)}{1 - p(C = +|x_1, ..., x_n)} = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n.$$

By taking the exponential of both sides the log can be removed to get

$$\frac{p(C = +|x_1, .., x_n)}{1 - p(C = +|x_1, ..., x_n)} = e^{\beta_0 + \beta_1 x_1 + ... + \beta_n x_n}$$

and $p(C = +|x_1, .., x_n)$ can then be found by simple algebra as

$$p(C = +|x_1, .., x_n) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n)}} = L(\beta_0 + \beta_1 x_1 + ... + \beta_n x_n),$$

where $L(x)$ is known as the logistic function, an S-shaped curve displayed in Figure 3.2 for a single feature $\boldsymbol{x} = x_1$, $\beta_0 = 0$ and $\beta_1 = 1$.

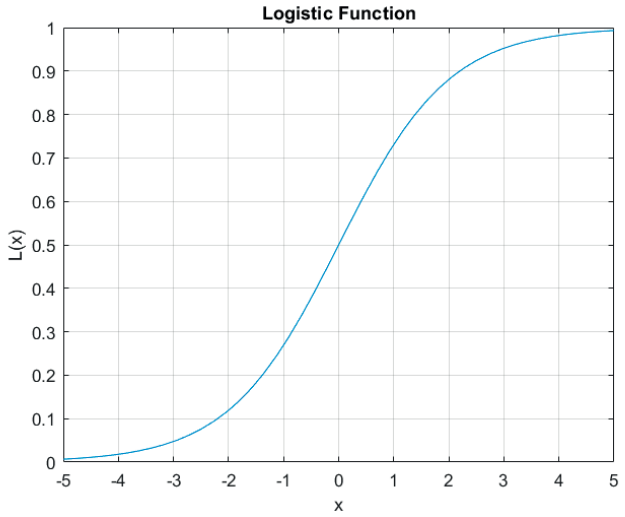

Figure 3.2: Logistic function

After the model is trained the parameters $\beta_0, ..., \beta_n$ will be set to a constant value. From there it is straight forward to calculate the probability $p(C = +|x_1, ..., x_n)$ which can then be used to classify the sample $x_1, ..., x_n$ to belong to the class $C = +$ if the probability is larger than or equal to 0.5. If the probability is smaller than 0.5 it will be classified as $C = -$.

3.7 Support Vector machine

Support vector machines are supervised learning ML models that can be used to solve both classification and regression problems. The support vector machine works by constructing a hyperplane, or set of hyperplanes, in a high (maybe infinite) dimensional space. The hyperplanes of a good classifier has a large distance (large margin) to the closest samples of any class, in order to properly separate the classes and achieve a low generalization error.

Provided some training vectors $\boldsymbol{x}_i \in \mathbb{R}^p, i = 1, ..., n$, each vector belonging to one of two classes represented by $\boldsymbol{y} \in \{1, -1\}^n$. The goal of the classifier is to find a $\boldsymbol{w} \in \mathbb{R}^p$ and a $b \in \mathbb{R}$ such that the output of $sign(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}) + b)$ is correct for as many samples as possible. The following primal problem is solved by the support vector classifier

$$\min_{\boldsymbol{w}, b, \boldsymbol{\zeta}} \frac{1}{2} \boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{n} \zeta_i,$$

$$y_i(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b) \geq 1 - \zeta_i,$$

$$\zeta_i \geq 0.$$

In an ideal dataset, the value of $y_i(\boldsymbol{w}^T \boldsymbol{\phi}(\boldsymbol{x}_i) + b)$ would be ≥ 1 for each sample, which would create a perfect classifier. In most real datasets classes are not completely separable by a hyperplane, the support vector classifier therefore allows some samples to be at a distance ζ_i from the exact margin. C in the equation above controls how much the model is penalized when a sample is incorrectly classified and can therefore act as an inverse regularization parameter.

The dual to the primal problem is

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \boldsymbol{\alpha}^T Q \boldsymbol{\alpha} - \boldsymbol{e}^T \boldsymbol{\alpha},$$

$$\boldsymbol{y}^T \boldsymbol{\alpha} = 0,$$

$$0 \leq \alpha_i \leq C,$$

where Q is a $n \times n$ positive semidefinite matrix defined as $Q_{ij} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, where $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^T \boldsymbol{\phi}(\boldsymbol{x}_j)$ is known as the kernel function. In this thesis the radial basis function kernel was used, which means that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}$ where $\gamma \geq 0$. The terms α_i are known as the dual coefficients and they can not be greater than C. The dual representation of the problem better illustrates that the feature vectors are mapped into a higher dimensional space by the $\boldsymbol{\phi}$ function.

When the optimization problem has been solved, the predicted class for a feature vector $\boldsymbol{x}$ will be the sign of

$$\sum_{i \in SV} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b.$$

It is sufficient to only sum over the support vectors since the dual coefficients α_i are zero for all the other samples.

3.8 Naive Bayes Classifier

Naive Bayes classifiers are supervised learning algorithms that are applying Bayes' theorem with the assumption that there is a conditional independence between each and every pair of features given the value of the class variable.

Bayes' rule states that the probability that a set of features, $x_1, ..., x_n$, belongs to a class c is

$$p(c|x_1, ..., x_n) = \frac{p(x_1, ..., x_n|c)p(c)}{p(x_1, ..., x_n)}.$$

In a binary classification problem, the feature set $x_1, ..., x_n$ will be classified as class $C = +$ when

$$f_b(x_1, ..., x_n) = \frac{p(C = +|x_1, ..., x_n)}{p(C = -|x_1, ..., x_n)} \geq 1,$$

where f_b is known as the Bayesian classifier.

Given the naive assumption that all features are conditionally independent the given class variable is

$$p(x_1, ..., x_n|c) = \prod_{i=1}^{n} p(x_i|c).$$

The resulting naive Bayesian classifier is then

$$f_{nb} = \frac{p(C = +)}{p(C = -)} \prod_{i=1}^{n} \frac{p(x_i|C = +)}{p(x_i|C = -)}.$$

Different naive Bayes classifiers mainly differ in how the distribution of $p(x_i|c)$ is assumed. For a Gaussian naive Bayes classifier, which was used in this thesis, the probability of the features conditioned a given class c is assumed to follow a Gaussian distribution

$$p(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2})},$$

where the mean, μ_c, and standard deviation, σ_c^2, is estimated using the maximum likelihood estimator.

3.9 Cross-Validation

Cross-validation is a simple and widely used technique for model validation and tuning of model hyperparameters. Cross-validation can be done in several ways and in this thesis K-fold cross-validation was used. In K-fold cross-validation the data is split into K folds (or groups) of equal size. Figure 3.3 gives an illustration of 5-fold cross-validation where, in each iteration, four folds are used to train the model and then the last fold is used to test the model, resulting in an accuracy for that iteration. The total accuracy is then the mean of the five calculated accuracies from the 5 iterations.

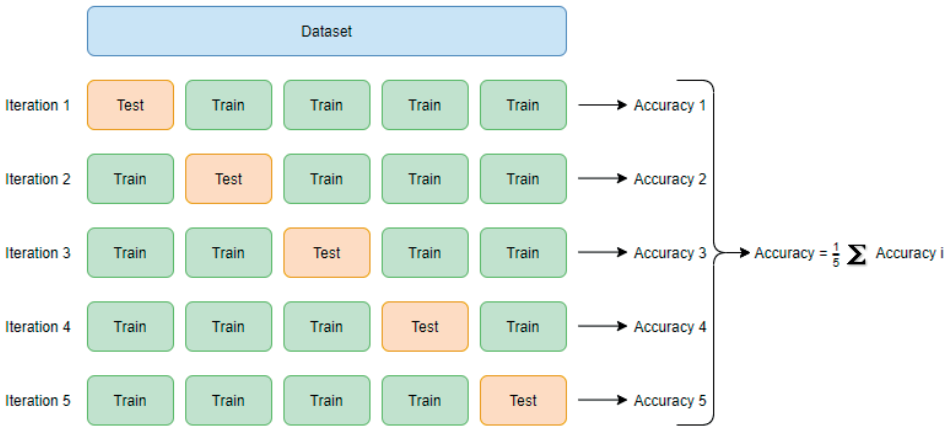

Figure 3.3: 5-fold cross-validation

Chapter 4
Simulation and Data Overview

This chapter explains and discusses how the simulator and ML was used to collect the results. The first and second section provides an overview of the simulation scenarios and how the simulator was used to generate the data set. The third section gives an overview of how the generated data was labeled and gives an explanation of the used features. The fourth section goes through how the was used train the ML models.

4.1 Simulation Overview

In this thesis the training data for the ML model was generated in an Ericsson LTE simulator that offered great flexibility in how the simulations could be performed and what data that could be logged. The simulator parameters used are summarized in the next section.

In the simulator it was possible to alter the network deployment by changing the number of sites and the number of sectors at each site. By adding a specified number of micro cells in each sector in the simulation area it was possible to use a heterogeneous network deployment. In order to not make the simulations too complicated, it was decided to run all the simulations with the same network deployment. The simulations all used one site with three sectors, resulting in a total of three cells in the simulated network.

There was several different user models available for use in the simulator. The user model that was used in this thesis was full-buffer users. The full-buffer user model is a simplistic model where the buffers of the users data flows always has unlimited data to transmit [13]. Each simulation was run with a fixed number of users for the entire simulation time, and the users where generated at random locations in the simulation area. In order to be able to reproduce simulation results, it was possible to select a seed when running a simulation. This means that the locations where the users are generated are determined by the selected seed. Figure 4.1 and Figure 4.2 gives an example of how the location of the generated users differ in a 5-user scenario generated with two different seeds. The simulations were run using 25 different seeds in order to capture several different user location scenarios.

In order to be able to compare the difference in downlink SINR when using different search widths, each simulation scenario had to be run twice. One time

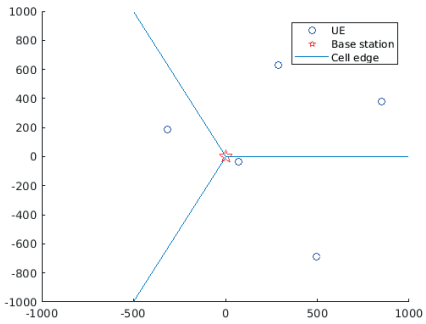

Figure 4.1: UE positions in seed 1

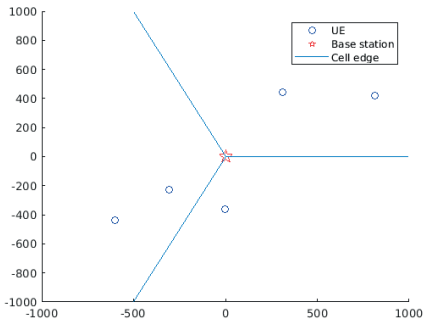

Figure 4.2: UE positions in seed 2

for each search width used in the dynamic point blanking algorithm. This means that the total number of simulations run was $30 \cdot 5 \cdot 25 \cdot 2 = 7500$ and the total number of samples for the data set was $\frac{7500}{2} = 3750$.

The next section provides an overview of all the relevant simulation parameters that was used.

4.2 Simulator Parameters

- Simulation time: 6.0 s
- Number of seeds: 25
- Deployment
  - Number of sites: 1
  - Number of sectors per site: 3
  - Cell radius: 577 m
- System frequency
  - Bandwidth: 9 MHz
  - Number of subbands: 50
  - Carrier frequency: 2.0 GHz
- Antenna
  - Number of antenna elements: 2
  - Electrical antenna tilt: Simulated with 0, 3, 6, 9 and 12 degree downtilt
- UEs
  - Full-buffer user model

  - Number of UEs: Fixed for each simulation and simulated with 1-30 UEs
  - Movement: Straight mover with 0.83 m/s velocity (pedestrian)
- DPB search width: Each scenario was simulated with two different search widths. One simulation using DPB search width two and one simulation using full DPB search width.

4.3 Data Overview

In order for the ML model to make correct decisions in a given environment, it first needs to learn from some training data. The data set has a great impact on the performance of the ML model, it is not possible to get satisfying results if the structure of the data set is not chosen with great care. In this thesis it was decided to go for a supervised learning approach, which means that it had to be decided what the set of input features should be and how the data should be labeled.

4.3.1 Labeling the data

The first step of deciding how to label the data was to decide what type of key performance indicator (KPI) to use. Since the purpose of the thesis was to reduce the inter-cell interference, it was decided to use the mean SINR of the UEs as the KPI.

One problem with labeling of data is that using full search basically always gets the highest SINR. This means that the correct decision can not be the search width that gets the highest SINR, since this would lead to full search being the correct decision for all scenarios. One way to label the data is to include a feature that measures the available processing power, and select a smaller search width if there is not much available processing. Unfortunately, there was no direct way of accessing the available processing power in the used simulator therefore this way of labeling the data was not used. It was decided to use a threshold value as the basis for labeling the data. The threshold value was set to be 95 % of the SINR that was gained by full search. To clarify: If search width two resulted in more than 95 % of the SINR of full search width, then the best decision was to use a search width equal to two. In Figure 4.3 it can be seen that the correct decision is search width two when there are 15, 16 and 17 users in the system. With 18, 19 and 20 users, full search is the best decision.

By studying the results of the simulator logs, it could be observed that the difference in SINR between using two search width and full search width was in general not very large. It was therefore decided to use a binary classifier ML model. This makes the system easier to model and therefore makes the ML model less complex.

4.3.2 Feature Selection

When it had been determined how to label the data and which KPI to use, the next step was to chose a set of input features to the ML model. The features must

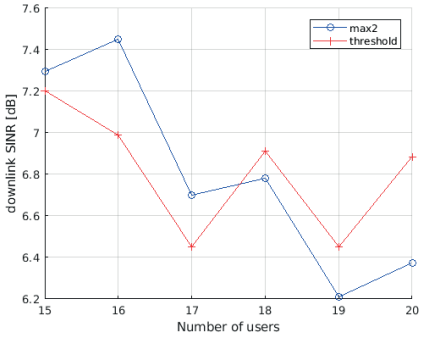

Figure 4.3: The figure shows the resulting SINR of using search width two and the threshold 95 % of full search width

have an impact on the resulting label or else it is not possible for the model to learn anything from them and they do not add any valuable information. In this case "impact on the resulting label" means that the feature has an effect on the SINR difference between search width two and full search width. The approach to finding features was that it was thought that in scenarios where there is more inter-cell interference the difference will be larger compared to scenarios where the inter-cell interference is lower. It was not obvious what features to choose and the majority of the thesis work was spent on trying to find suitable ones, since this is crucial for the performance of the model. The features that was used in this thesis are described in the following sections.

Number of Users

The total number of users is the number of users in the simulation area. The simulations where run with a fixed number of users and it was therefore easy to extract the number of users used in a certain simulation. It was believed that the number of users would have an impact on the inter-cell interference, since when there are more users there is an increased risk that at least one user is experiencing inter-cell interference. When there are only one or a few users in the area the probability is lower, since it is probable that not all cells serves any users. It can also be seen in Figure 4.3 that the correct decision changes depending on the number of users in the system.

Electrical Antenna Tilt

Electrical antenna tilt is a way to control the base station antenna downtilt, that is, the angle of the antennas radiating beam relative to the horizontal plane [12], see Figure 4.4.

This feature has a large impact on the network performance. A larger downtilt of the antenna results in a beam that is more concentrated to users that are closer to the base station and could therefore have an impact on the users experienced SINR. Figure 4.5 shows the impact electrical tilt can have on how the data is

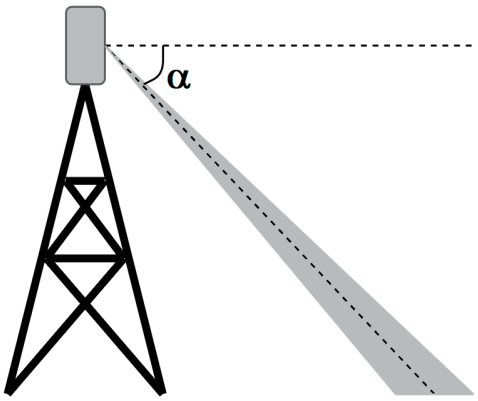

Figure 4.4: Illustration of the electrical antenna tilt α

labeled. In this scenario, using search width two is better when the tilt is 0, 3 and 6 degrees. Full search width is better when the tilt is 9, 12 and 15 degrees.

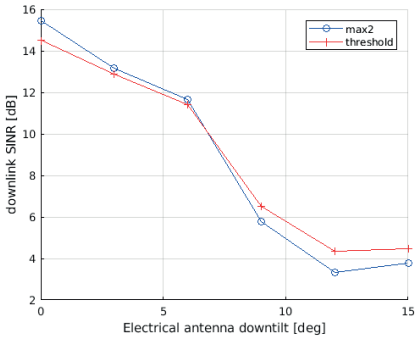

Figure 4.5: The figure shows SINR vs. antenna tilt for a ten user scenario

User Coordinates

The user coordinates gives the positions of the users in the simulation area. The idea behind using this data was to be able to calculate the number of users that are within a certain distance from the cell edges. The users that are close to the cell edges are the ones that are most likely to experience large inter-cell interference and therefore are expected to gain the most from dynamic point blanking. The number of users within 200 m of a cell edge was used as a feature of the ML model, see Figure 4.6. If a user was inside of the dashed line it was considered a cell edge user.

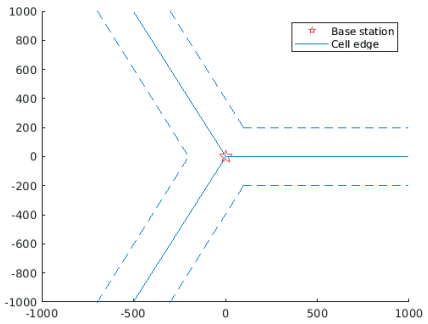

Figure 4.6: Display of the area where UEs were considered to be edge UEs

Cell Utilization

The cell utilization is the number of active PRBs divided by the total number of PRBs in a cell. The idea behind using this as a feature is that it could reflect how the load is distributed in the simulation area. One problem that arises when using this as a feature is that using different search widths results in a different cell utilization. The way that this problem was mitigated in this thesis was by always taking the cell utilization when using search width two. If implemented, this would mean that there would have to be periods when the ML is turned off and search width two is used, in order to measure the cell utilization in the same environment that the ML has trained on. The cell utilization gave three features, since there were three cells used in the simulations.

User Channel Quality Hypotheses

The user channel quality hypotheses is the channel quality hypotheses that is estimated from the CSI measurements. This gives the estimated SINR gain when muting zero, one or two cells for each UE in the simulation area. The idea behind using this as a feature is that if the users are expected to gain a lot in SINR if neighboring cells are muted, then there should be much inter-cell interference. Both the mean estimated SINR gain, lower and upper quartile gain was included in the generated data set, giving three additional features to the data set.

4.4 Post Processing and Result Collecting

After the dataset had been generated from the raw output logs from the simulations it was time to implement the three chosen ML models. This was done using the scikit-learn python library [14].

Scaling

The dataset was scaled so that each feature was in the range $[-1, +1]$. Scaling is very important to do when using certain models. The main reason for scaling the data is to avoid that some features that has a greater numeric range dominates the features that does not vary as much. In the used dataset the number of users was in the range $[1, 30]$ and can be compared to the cell utilization that was in the range $[0, 1]$. If scaling was not done there is a risk that the the number of users feature would take the upper hand and reduce the performance of the models. Another reason to scale the dataset is that is reduces the numerical difficulties during the calculations.

Train-Test Split

The dataset was separated into a training set and a test set. The training set consisted of two thirds of the samples and the test set consisted of the remaining third of the samples. The models where first validated by doing 5-fold cross-validation on the training samples and then trained using the data samples in the training set. The result where collected by letting the models predict the outcome of the testing samples.

Cross-Validation

The performance of all the used models where validated using stratified 5-fold cross-validation on the training set. 5-fold cross-validation was also used for optimizing the hyperparameters γ and C for the support vector classifier. The values used for the hyperparameters were found by doing a grid search. At each point in the grid the performance of the support vector classifier was measured by doing cross-validation. The values chosen for C and γ were then at the point in the grid where the cross-validation yielded the best result.

Chapter 5
Results

In this chapter all the results of the thesis are presented. The chapter is divided into five sections. The first section presents the statistics of how the label was impacted by the different used features. The second section gives some background to the results and explains the metrics that are presented in the tables and figures. Section three to five present the results of the three evaluated machine learning models.

5.1 Dataset Statistics

The figures in this section all display how the mean label is impacted by a change in the features. The mean label is in this binary case a value between zero and one. A value of the mean label that is smaller than 0.5 means that most samples at this point are labeled as two search and values above 0.5 are mostly labeled as full search. A value of 0.5 means that it is difficult to determine the label.

Figure 5.1 shows how the mean label is impacted by the number of users used in the simulation. It is clear that the number of users has an impact on the label and having fewer users in the system leads to two search being selected more often. More users in the system means that full search is the most common label.

Figure 5.2 shows a similar pattern as the one shown in Figure 5.1. A lower cell utilization generally means that two search is the best option to use. A higher cell utilization leads to full search being the best search width to use.

The pattern in Figure 5.3 is not as clear as in the previous figures. The mean label is relatively close to 0.5 for all values of the antenna tilt except for when the tilt is nine degrees.

The number of edge users versus the mean label is shown in Figure 5.4. It can be seen that the mean label is not clearly impacted by this feature. The mean labels goes from 0.47 for zero edge users up to 0.72 for nine edge users. This feature is also correlated with the number of users feature and may therefore not add much information to the ML model. The muting gain hypothesis feature versus the mean label is shown in Figure 5.5. The mean label goes from 0.75 when the muting gain is equal to ten down to 0.5 when it is equal to 25.

When the results were being extracted it was discovered that the number of cell edge users feature and the user channel quality hypotheses features did not improve the performance of the ML algorithms in any regard. These features are therefore

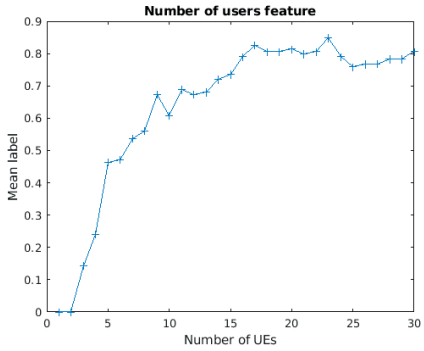

Figure 5.1: Number of users feature

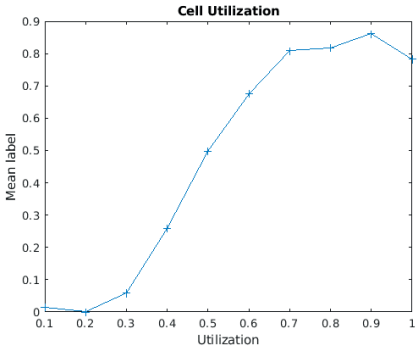

Figure 5.2: Cell utilization feature

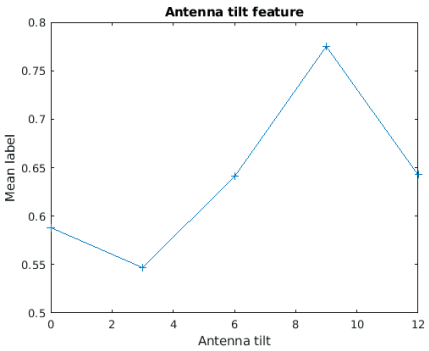

Figure 5.3: Electric antenna tilt feature

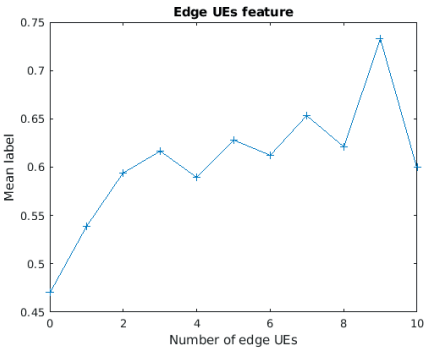

Figure 5.4: Number of edge users feature

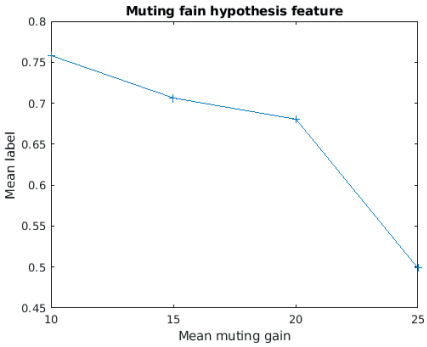

Figure 5.5: Muting gain hypothesis feature

not included in the data set that was used to generate the results presented in this chapter. The final used features was electrical antenna tilt, number of users, cell one utilization, cell two utilization and cell three utilization.

5.2 Performance Metrics

The results are shown as three different metrics. The metrics are the precision, recall and F1-score.

The precision is the proportion predicted zero cases that are correctly real zeros. When given a general confusion matrix as in Figure 5.6, the precision can be calculated as

$$A/(A+B),$$

where A is the number of correctly predicted zero cases and B is the number of zero samples that was incorrectly predicted as one.

The recall score is the proportion of real zero cases that are correctly predicted zero. In Figure 5.6 the recall can be calculated as

$$A/(A+C),$$

where A is the number of correctly predicted zero cases and C is the number of samples that was incorrectly predicted as zero.

The F1-score is the harmonic mean of the precision and recall.

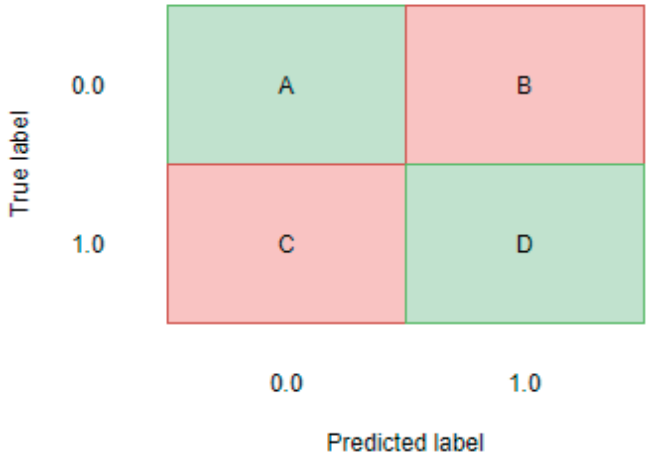

Figure 5.6: An illustration of a binary confusion matrix

5.3 Logistic Regression

The resulting metrics for the logistic regression model is shown in Table 5.1. The 0 and 1 in the two first rows in the leftmost column is the label for search width

two and full search width, respectively. The support column shows that there was a total of 1250 samples in the test set. Of these 1250 samples, 455 was labeled with two search width and 795 was labeled with full search width.

	Precision	Recall	F1-score	Support
0	0.86	0.74	0.80	455
1	0.86	0.93	0.89	795
Accuracy	-	-	0.86	1250
Macro avg	0.86	0.84	0.84	1250
Weighted average	0.86	0.86	0.86	1250

Table 5.1: Logistic regression metrics

It can be seen that the precision is 86 % for both labels. The recall metric is 74 % for the two search width and 93 % for the full search width label, resulting in a macro average of 84 % and a weighted average of 86 %. The F1-score follows the pattern of the recall score with two search having a 80 % F1-score and full search having 89 %. Figure 5.7 shows the resulting confusion matrix of the logistic regression model.

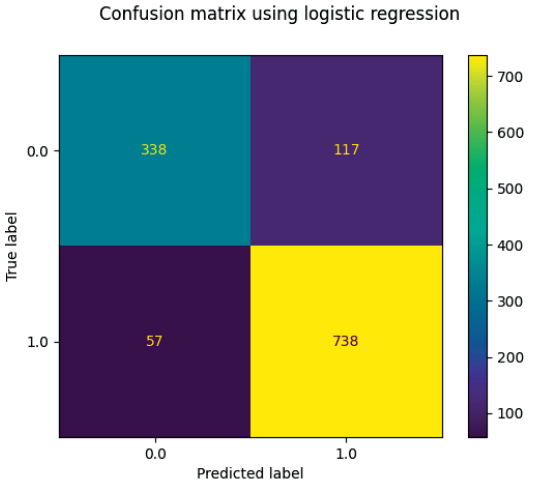

Figure 5.7: Logistic regression confusion matrix

5.4 Support Vector Classifier

Figure 5.8 shows the results of the grid search that was performed in order to tune the hyperparameters C and γ for the support vector classifier. It can be seen that the best validation accuracy is achieved when $C = 1.0$ and $\gamma = 1.0$ or when $C = 100.0$ and $\gamma = 0.1$. The values used for the final model was $C = 1.0$ and $\gamma = 1.0$.

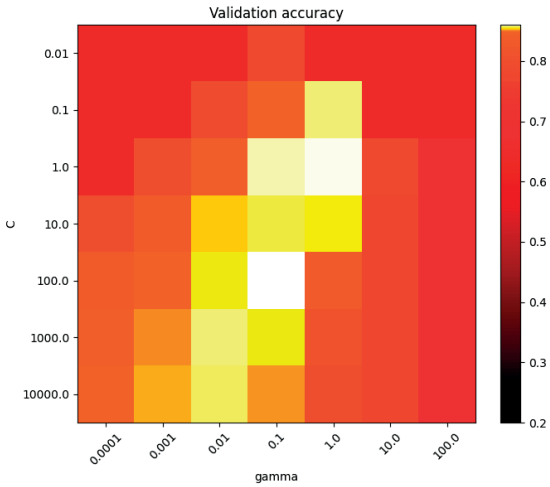

Figure 5.8: Support vector classifier grid search

The resulting metrics for the support vector classifier is shown in Table 5.2. The 0 and 1 in the two first rows in the leftmost column is the label for search width two and full search width, respectively. The support column shows that there was a total of 1250 samples in the test set. Of these 1250 samples, 455 was labeled with two search width and 795 was labeled with full search width.

	Precision	Recall	F1-score	Support
0	0.90	0.77	0.83	455
1	0.88	0.95	0.91	795
Accuracy	-	-	0.89	1250
Macro avg	0.89	0.86	0.87	1250
Weighted average	0.89	0.89	0.88	1250

Table 5.2: Support vector classifier metrics

It can be seen that the precision is 90 % for the two search width label and 88 % for the full search label. The recall metric is 77 % for the two search width and 95 % for the full search width label, resulting in a macro average of 86 % and a weighted average of 89 %. The F1-score is 83 % for two search and 91 % for full search. Figure 5.9 shows the resulting confusion matrix of the support vector classifier.

5.5 Naive Bayes

The resulting metrics for the naive Bayes classifier is shown in Table 5.3. The 0 and 1 in the two first rows in the leftmost column is the label for search width two and full search width, respectively. The support column shows that there was a

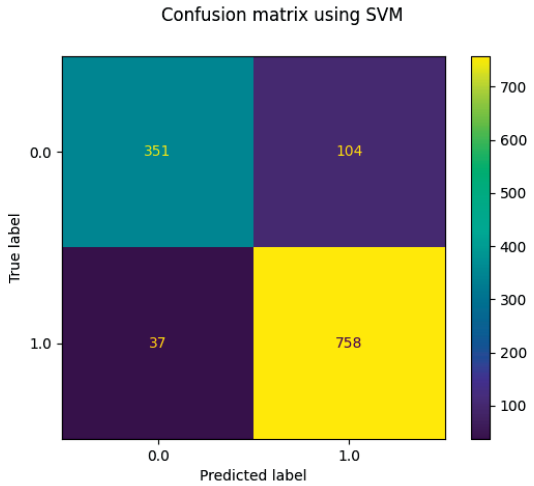

Figure 5.9: Support vector classifier confusion matrix

total of 1250 samples in the test set. Of these 1250 samples, 455 was labeled with two search width and 795 was labeled with full search width.

	Precision	Recall	F1-score	Support
0	0.88	0.67	0.76	455
1	0.83	0.95	0.89	795
Accuracy	-	-	0.85	1250
Macro avg	0.86	0.81	0.82	1250
Weighted average	0.85	0.85	0.84	1250

Table 5.3: Naive Bayes metrics

It can be seen that the precision is 88 % for the two search width label and 83 % for the full search label. The recall score is much better for the full search label with a score of 95 % compared to a score of 67 % for the two search label. The F1-score is 76 % for two search and 89 % for full search. Figure 5.10 shows the resulting confusion matrix of the naive Bayes model.

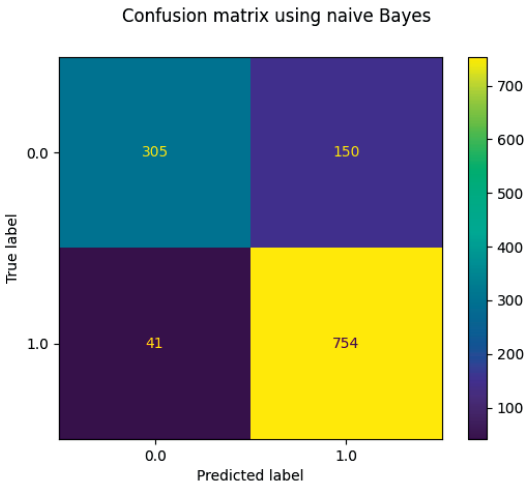

Figure 5.10: Naive Bayes confusion matrix

Chapter 6
Discussion

6.1 Overall Results

The results show that the support vector classifier was the model that achieved the best results out of the three models that was evaluated. The reason for this result is probably because of nonlinearities present in the collected dataset. The logistic regression model is a linear model and may therefore not be able to reach the same results that the support vector classifier does. The naive Bayes classifier is the models that produces the worst results out of the three. This could be because of the models assumption that there is a conditional independence between the input features given a certain label, which may be too far from the truth in this specific dataset.

Another interesting thing to discuss about the results is that all models had a harder time predicting the label when a sample was labeled as two search compared to full search. The straightforward explanation for this is that the dataset was a bit imbalanced, meaning that there was an uneven distribution between the labels. Training a model with an unbalanced dataset naturally leads to fewer training samples of the minority label and it can therefore be hard for the model to correctly identify the scenarios where a sample should be mapped to the minority label.

One thing that possibly could slightly improve the ability of the models to correctly identify a two search scenario is to increase the total number of samples in the dataset. If there are more samples then the number of minority labeled samples would also increase, leading to a better possibility for the models to identify the characteristics of the two search scenarios. During the intermediate phase of the model evaluation, it was explored how the results varied with different sizes of the train/test-split of the dataset. It was noted that there was no substantial increase in the performance when using a 50/50 split compared to a 66/33 split and it is therefore unlikely that increasing the amount of samples in the dataset would be very beneficial.

Generating a new dataset with a not as uneven distribution of the class labels would be another way of getting a better result. This could be done by lowering the SINR threshold that the two search has to pass in order to be selected as the correct label. By lowering the threshold, the amount of samples that would be labeled as the two search class would increase and it is likely that a threshold between 0.90 and 0.95 would result in a more balanced dataset.

6.2 Future Work

In order for the machine learning approach used for dynamic point blanking in this thesis to be beneficial, it is necessary to know how expensive it is to evaluate the muting pattern hypotheses. The full search will in the majority of situations result in a better SINR and the only reason to lower the search width is to save processing power. If it is not known how much the gain in processing capacity is by using two search then it is impossible to quantify the gain in lowering the search width. This has to be explored further before it is worthwhile to use in a real world scenario.

The scope of this thesis was limited to full buffer users in a one site with three sectors network. In future work it could be studied how the inter-cell interference is impacted by different kinds of user models and deployment scenarios. The full buffer user model is not the most realistic user model and it could be beneficial to use other user models in order to get a better view of how the behaviour would be in a real world deployment. It would also be interesting to study the behaviour in heterogeneous networks, since these kinds of deployments are another scenario where inter-cell interference often is present.

An alternate way of using machine learning to implement dynamic point blanking could be to discard the search tree and instead let the machine learning model directly select the muting pattern. This approach would completely eliminate the need of evaluating muting hypotheses and could therefore potentially save a lot of processing power. The drawback with going for this approach would be that, because of the large number of possible muting patterns, there would be a very large number of classes for the model to choose from. Having such a large number of classes to choose from would require a sophisticated and complex machine learning model which in itself could require more processing power and memory than the muting hypothesis evaluation.

References

[1] A. F. Molisch, "Orthogonal Frequency Division Multiplexing (OFDM)" in *Wireless Communications*, 2nd ed. Wiley, 2011, pp. 417

[2] E. Dahlman, S. Parkvall, J. Sköld, *5G NR: The Next Generation Wireless Access Technology.*, Elsevier, 2018

[3] E. Dahlman, S. Parkvall, J. Sköld, "Multi-Point Coordination and Transmission" in *4G, LTE-Advanced Pro and The Road to 5G*, 3rd ed. Academic Press, 2016, pp. 331-345

[4] J. Hu, *et al.* "Voronoi-Based Multi-Robot Autonomous Exploration in Unknown Environments via Deep Reinforcement Learning" in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14413-14423, 2020

[5] G. Rebala, A. Ravi, S. Churiwala, "Learning Models" in *An Introduction to Machine Learning*, Springer, 2019

[6] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*, 2nd ed. MIT press, 2018

[7] T. Hastie, *et al. The Elements of Statistical Learning. Data Mining, Inference, and Prediction.*, 2nd ed. Springer, 2009

[8] C. M. Bishop, *Pattern recognition and machine learning.*, 2nd ed. Springer, 2006

[9] C. Cortes, V. Vapnik *Support-vector networks*, Machine Learning, vol. 20, no. 3, pp. 273–297, 1995

[10] AJ. Smola, B. Scholkopf *A tutorial on support vector regression*, Statistics and Computing, 2004

[11] Zhang, Harry, *The Optimality of Naive Bayes*, American Association for Artificial Intelligence, 2004

[12] Vanella, F. *et al. Remote Electrical Tilt Optimization via Safe Reinforcement Learning*, IEEE Wireless Communications and Networking Conference (WCNC), 2021

[13] P. Ameigeiras, *et al. Traffic models impact on OFDMA scheduling design*, EURASIP Journal on Wireless Communications and Networking, 2012

[14] F. Pedregosa, *et al. Scikit-learn: Machine Learning in Python*, JMLR 12, 2011, pp. 2825-2830

Printed by Tryckeriet i E-huset, Lund 2021

LUND
UNIVERSITY

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2021-847
http://www.eit.lth.se