Master of Science Thesis in Applied Mathematics
Department of Electrical Engineering, Linköping University, 2018

5G Positioning using Machine Learning

Magnus Malmström

Master of Science Thesis in Applied Mathematics

5G Positioning using Machine Learning

Magnus Malmström

LiTH-ISY-EX--18/5124--SE

Supervisors: **Yuxin Zhao**
ISY, Linköping University
Sara Modarres Razavi
Ericsson Research Linköping
Fredrik Gunnarsson
Ericsson Research Linköping

Examiner: **Isaac Skog**
ISY, Linköping University

Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

To my family and friends.

Sammanfattning

Radiobasserad positionering av användarenheter är en viktig applikation i femte generationens (5G) radionätverk, som mycket tid och pengar läggs på för att utveckla och förbättra. Ett exempel på tillämpningsområde är positionering av nödsamtal, där ska användarenheten kunna positioneras med en noggrannhet på ett tiotal meter. Radio basserad positionering har alltid varit utmanande i stadsmiljöer där höga hus skymmer och reflekterar signalen mellan användarenheten och basstationen. En ide att positionera i dessa utmanande stadsmiljöer är att använda datadrivna modeller tränade av algoritmer baserat på positionerat testdata – så kallade maskininlärningsalgoritmer.

I detta arbete har två icke-linjära modeller - neurala nätverk och random forest – bli implementerade och utvärderade för positionering av användarenheter där signalen från basstationen är skymd.Utvärderingen har gjorts på data insamlad av Ericsson från ett 5G-prototypnätverk lokaliserat i Kista, Stockholm. Antennen i den basstation som används har 48 lober vilka ligger i fem olika vertikala lager. Insignal och målvärdena till maskininlärningsalgoritmerna är signals styrkan för varje stråle (BRSRP), respektive givna GPS-positioner för användarenheten. Resultatet visar att med dessa maskininlärningsalgoritmer positioneras användarenheten med en osäkerhet mindre än tio meter i 80 procent av försöksfallen.

För att kunna uppnå dessa resultat är viktigt att kunna detektera om signalen mellan användarenheten och basstationen är skymd eller ej. För att göra det har ett statistiskt test blivit implementerat. Detektionssannolikhet för testet är över 90 procent, samtidigt som sannolikhet att få falskt alarm endast är ett fåtal procent.

För att minska osäkerheten i positioneringen har undersökningar gjorts där utsignalen från maskininlärningsalgoritmerna filtreras med ett Kalman-filter. Resultat från dessa undersökningar visar att Kalman-filtret kan förbättra presitionen för positioneringen märkvärt.

Abstract

Positioning is recognized as an important feature of fifth generation (5G) cellular networks due to the massive number of commercial use cases that would benefit from access to position information. Radio based positioning has always been a challenging task in urban canyons where buildings block and reflect the radio signal, causing multipath propagation and non-line-of-sight (NLoS) signal conditions. One approach to handle NLoS is to use data-driven methods such as machine learning algorithms on beam-based data, where a training data set with positioned measurements are used to train a model that transforms measurements to position estimates.

The work is based on position and radio measurement data from a 5G testbed. The transmission point (TP) in the testbed has an antenna that have beams in both horizontal and vertical layers. The measurements are the beam reference signal received power (BRSRP) from the beams and the direction of departure (DoD) from the set of beams with the highest received signal strength (RSS). For modelling of the relation between measurements and positions, two non-linear models has been considered, these are neural network and random forest models. These non-linear models will be referred to as machine learning algorithms.

The machine learning algorithms are able to position the user equipment (UE) in NLoS regions with a horizontal positioning error of less than 10 meters in 80 percent of the test cases. The results also show that it is essential to combine information from beams from the different vertical antenna layers to be able to perform positioning with high accuracy during NLoS conditions. Further, the tests show that the data must be separated into line-of-sight (LoS) and NLoS data before the training of the machine learning algorithms to achieve good positioning performance under both LoS and NLoS conditions. Therefore, a generalized likelihood ratio test (GLRT) to classify data originating from LoS or NLoS conditions, has been developed. The probability of detection (P_D) of the algorithms is about 90% when the probability of false alarm (P_{FA}) is only 5%.

To boost the position accuracy of from the machine learning algorithms, a Kalman filter have been developed with the output from the machine learning algorithms as input. Results show that this can improve the position accuracy in NLoS scenarios significantly.

Acknowledgments

This thesis concludes my education in Applied Physics and Electrical Engineering at Linköping University. These five years have truly been great years.

First and foremost, I would like to thank Ericsson Research Linköping, LINLAB, for given me the opportunity to write this challenging and interesting thesis. A special thanks to my two supervisors at Ericsson, Sara Modarres Razavi and Fredrik Gunnarsson, for taking their time to answer all my questions and having a genuine interest in my thesis and results.

I would also like to thank my supervisor at Linköping University, Yuxin Zhao, and my examiner, Issac Skog, for providing new ideas and angles to my work. Without Yuxin, Sara, Fredrik and Isaac's keen eyes my report would not look as good and tidy as it now does.

I would also like to take the opportunity to thank all the people I have met during these years and all the friends I have made during my studies, friends I have studied with, friends who I have worked with in various student activities I have been part of, and of course my friends from my exchange studies in Eindhoven. It is all of you who have made these years some of the most fun and memorable years of my life.

Last but not least, I would like to thank my family for all the support they have given me during my studies. For this I will always be grateful. You have had a bigger part than you might think in the making of this thesis.

Linköping, May 2018
Magnus Malmström

Contents

Notation

Abbreviation

Abbreviation	Definition
3GPP	3rd Generation Partnership Project
5G	5th Generation cellular networks
BLE	Bluetooth Low Energy
BRSRP	Beam Reference Signal Received Power
CART	Classification And Regression Trees
CDF	Cumulative Distribution Function
CNN	Convolutional Neural Network
CRLB	Cramér-Rao Lower Bound
DC	Direct Current
DoD	Direction of Departure
ECDF	Empirical Cumulative Distribution Function
GPS	Global Positioning System
GLRT	Generalized Likelihood Ratio Test
LoS	Line of Sight
LTE	Long Term Evolution
MAP	Maximum A Posteriori
MLE	Maximum Likelihood Estimator
MUSIC	MUltiple SIgnal Classification
NLoS	Non Line of Sight
NP	Neyman-Pearson
OoB	Out of Bag
PDF	Probability Density Function
RSS	Received Signal Strength
TA	Timing Advance
TP	Transmission Point
UE	User Equipment
WGN	White Gaussian Noise

Defined parameters

Notation	Definition
P_D	Probability of detection
P_{FA}	Probability of false alarm
$\mathcal{L}$	Learning set
$\mathcal{D}$	Target set
$\mathcal{H}_0$	Null hypothesis
$\mathcal{H}_1$	Alternative hypothesis

Mathematical notation

Notation	Meaning
$\mathbf{x}$	The vector $\mathbf{x} = (x_1, \cdots, x_n)^T$
$\|\| \cdot \|\|_2^2$	The squared Euclidean norm
$[\ \cdot\ ]^T$	Vector/matrix transpose
$[\ \cdot\ ]^{-1}$	Matrix inverse
$\hat{x}$	The estimate of the variable x
$\ln(\ \cdot\ )$	Natural logarithm
$\nabla_{\mathbf{x}}$	Gradient with respect to the vector of variables $\mathbf{x}$
$\mathcal{N}(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$\xrightarrow{a.c.}$	Asymptotic convergence
$\mathbf{I}$	Identity matrix
$\mathbf{1}_{X_i \leq t}$	Indication function for the event $X_i \leq t$
$\mathbb{E}(x)$	Expected value of the random variable x

1

Introduction

This chapter introduces the problem formulation and the purpose of the thesis. The background work and the scenarios that are investigated are also elaborated.

1.1 Background

Positioning is recognized as an important application for the long-term evolution (LTE) and fifth generation (5G) cellular networks. This is due to its potential for massive commercial use cases e.g., industry automation, remote operation, and emergency call-outs. But also because of regulations in the United States, where it since October 2001 has been mandatory to have location-based services for the local network operators [1]. Radio based positioning has always been a challenging task in urban environments, where high-rising buildings block and reflect the signal between the user equipment (UE) and the transmission point (TP). These environments will be referred to as *urban canyons*. A schematic illustration of the positioning of two UEs in an urban canyon is shown in Figure 1.1.

In Figure 1.1 the first UE (green check mark) is said to be in line of sight (LoS), that is when there is a direct and clear path between a UE and the TP. While when the sight is blocked by several high-rising buildings as it is for the second UE (red cross) in Figure 1.1, the UE is in non line of sight (NLoS). In LoS there are many positioning methods that can be used to position the UE with high accuracy. It is even possible to obtain high positioning accuracy for UE travelling up to 100 km/h [2]. Here the timing advance (TA) i.e., the propagation delay of the signal between the UE and TP, and the direction of departure (DoD) between the beam with strongest beam reference signal received power (BRSRP) on the TP and a UE are used to estimate the position. An example of an algorithm to estimate DoD is multiple signal classification (MUSIC) [3]. Geometric LoS positioning methods

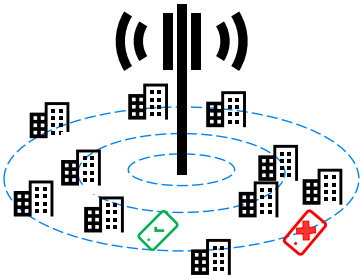
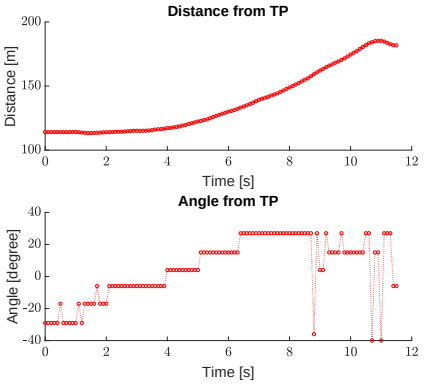


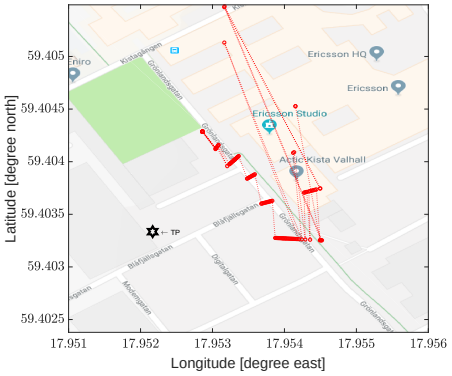
Figure 1.1: *Schematic picture of positioning in an urban canyon.*

will here be referred to as traditional positioning methods. Figure 1.2 shows an example of a traditional positioning method where the estimated DoD and distance are used for positioning a UE in an urban canyon. In this example the distance between the UE and TP is assumed known. In Figure 1.2a the estimated DoD and the true distance are shown, and in Figure 1.2b they are combined for positioning of a UE.

When the UE in Figure 1.2b gets behind a building (passes the crossing between Blåfjällsgatan and Grönlandsgatan) and enters a region with NLoS conditions, the estimated position of the UE (displayed as red points) is not reliable. This can also be seen in Figure 1.2a where the angle estimate fluctuates a lot after 8 seconds.



(a) *Upper: Target distance from TP calculated using GPS coordinates. Lower: Target angle from TP calculated from DoD.*



(b) *Estimated position of UE in an urban canyon using estimated angle and known distance between UE and TP.*

Figure 1.2: *Positioning using direction of departure and distance measurements method. This is done for the three different path the UE have travelled in the testbed.*

This fluctuating in DoD makes the traditional positioning method perform poorly. This calls for new positioning technologies that can handle NLoS conditions.

One approach for positioning in NLoS is to use data driven methods, referred herein as machine learning algorithms [4]. Detection of NLoS is also an important feature, that makes it possible to decide when traditional positioning methods can and cannot be used.

The purpose of this master thesis is to investigate the use of machine learning algorithms to perform positioning in urban canyons with NLoS conditions. Focus will be on positioning using "snapshot data". That means measurements are taken at a single time instance at each UE location. Selection of input to the machine learning algorithms is central, as well as use of the antenna design and its beam pattern when creating these input features. This thesis will also consider detection of NLoS conditions, as well as investigate the use of filtering of the output of the machine learning algorithms to improve the positioning accuracy.

1.2 Problem Formulation

The thesis aims to answer the following questions:

1. Is positioning in urban canyons with NLoS conditions possible using machine learning algorithms? If it is, what is the expected positioning accuracy?
2. Can LoS and NLoS conditions be distinguished using features related to DoD and BRSRP?
3. Is it possible to improve the positioning accuracy by filtering the output of the machine learning algorithms?

To investigate these questions, 5G testbed data is used, i.e., data generated from an early prototype of a 5G cellular network with prototypes both of the TP and UE. The data is used both for training and evaluation of the machine learning algorithms for positioning, as well as for studying of the characteristics of BRSRP and DoD in NLoS conditions. The data originate from a test carried out in Kista, Stockholm, by Ericsson AB.

The signal from the antenna has a carrier frequency of 15GHz and the structure of the antenna is given in Figure 3.2. The antenna has 48 beams arranged in five different vertical layers. Based on the testbed data, it is assured that the UE always has connection with the TP. In the testbed, there is only one UE connected to the TP, hence positioning of multiple UEs in the network and interference between users are not considered. There is also a limitation on the amount of data available for learning and evaluation of the different machine learning algorithms. This thesis will not consider modelling the environment and what is causing the NLoS and the differences in the signal characteristics, such as whether it is a tree or a building blocking the signal between the UE and the TP. The effect of different weather conditions and environments will neither be considered in the thesis.

1.3 Related Work

Positioning in cellular networks is a well established research topic. Many attempts have been done using traditional methods that estimate the DoD and the TA. In [2] the positioning of a car travelling in high speed is done using estimation of DoD from a TP. Also [5] investigates estimation of angle of arrival and position in cellular networks under LoS conditions using received signal strength (RSS) from directional antennas. Here they manage to estimate the angle of arrival with three degrees precision, and position with a sub-meter precision. These methods work well in LoS conditions where the TA can be estimated. From this, the distance can be calculated by dividing the TA with the speed of the signal, i.e., speed of light.

Under NLoS condition these methods are no longer applicable, therefore experiments using data driven methods have been investigated. In [4] the machine learning method Gaussian processes is used for positioning in NLoS. This is done in an indoor environment using the RSS from Bluetooth low energy (BLE) beacons.

A statistical model approach for positioning of UEs in urban canyons with NLoS conditions have been applied in [1], where the positioning accuracy is less than 300 meters. Their test is being done for a city-scale environment while in this thesis study a neighbourhood scale test size is assumed. In a city-scaled environment, the goal is to position the UE in the right cell, while for positioning on a neighbourhood scale, the goal is to position the UE at an exact point. Using neural networks for positioning in urban canyons are described in [6]. They are using convolutional neural networks (CNN) in combination with fingerprinting-based positioning. In comparison, this thesis will investigate the use of Bayesian regularized artificial neural networks.

A theoretical foundation for detection of NLoS using a generalized likelihood ratio test (GLRT) is described in [7] where they are assuming that the variance in NLoS conditions is larger than in LoS conditions using TA as the signal for detection. This thesis takes inspiration from the theory presented in [7], while using RSS and DoD as the signals for detection of NLoS.

Filters are also commonly used for positioning and tracking of UEs in cellular networks. For example, in [8] and [9] positioning using particle filtering is described. In [10], the authors discuss possibilities and limitations of mobile positioning in cellular networks. One of the limitations are the difficulties of propositioning in NLoS conditions. Both model-based filtering and a sensor fusion approach are investigated in [10] with good results. In this thesis, filtering the output from the machine learning algorithms will be used for smoothing the position estimate.

2

Theoretical Background

This chapter will give the reader the necessary theoretical background to understand the topics covered in the thesis. The reader is assumed to have good prior knowledge of probability theory and statistical hypothesis testing. For complementary reading, the reader is referred to the references.

2.1 Machine Learning

In this section, two different machine learning algorithms are described, namely, feed-forward Bayesian regularized artificial neural networks, and random forests.

2.1.1 Neural networks

Neural networks are originally denoting attempts to find a mathematical representation of information processing in biological systems [11]. The idea is to let a network of neurons learn patterns during a learning phase so that it is possible for the network to classify new input data after the learning phase. In pattern recognition, a feed-forward neural network model (also known as multilayer perception model) is a powerful tool. The model is based on a linear combination of a fixed number of non-linear parametric functions, called basis functions. In turn, these basis function depend on the input parameters and adjustable weights. For a two-layer neural network the adjustable weights are denoted as

$$\mathbf{w} = w_{10}^{(1)}, \ldots, w_{MD}^{(1)}, w_{10}^{(2)}, \ldots, w_{KM}^{(2)}. \tag{2.1}$$

Adjustable weights in the form of $w_{ij}^{(n)}$, $j \neq 0$, are referred to as weights and $w_{i0}^{(n)}$ as biases. The weights are adjusted by optimizing a predetermined cost function

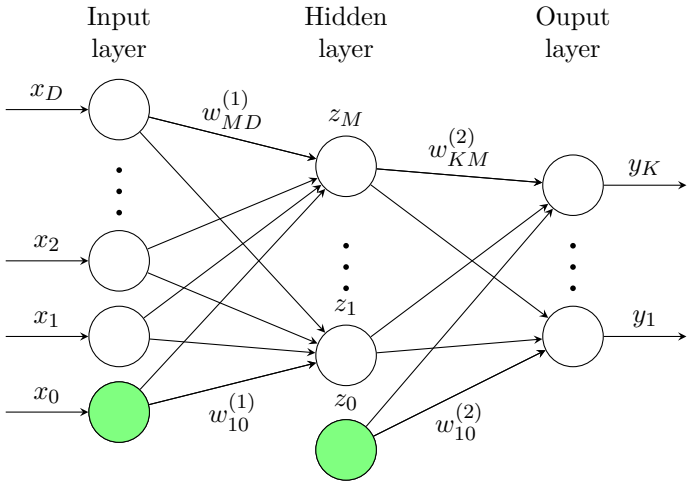


Figure 2.1: *Diagram of a two layer neural network. The input, output, and hidden layers are represented with vertices, and the weights with edges.*

$J(\mathbf{w})$. On a general form the two-layer neural networks is given by

$$y_k(\mathbf{x}, \mathbf{w}) = \nu\Big(\sum_{j=1}^{M} w_{jk}^{(2)} h\Big(\sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} x_0\Big) + w_{k0}^{(2)} z_0\Big), \tag{2.2}$$

where $\mathbf{x} = \{x_i\}$, $i = 1, \ldots, D$ is the input variable, and $\mathbf{y} = \{y_k\}_{k=1}^{K}$ is the output variable controlled by the vector $\mathbf{w}$ of adjustable weights given in (2.1). The parameters x_0 and z_0 are referred to as hidden variables [12], the hidden variables are often set to one. The superscript (1) stands for that the adjustable parameters are in the first layer, and (2) means that they are in the second layer. Transformation from the first to the second layer and from the second layer to the output are done by differentiable, non-linear activation functions $h(\,\cdot\,)$ and $\nu(\,\cdot\,)$, respectively. A commonly used activation function is the sigmoid function [12, 13]. That is,

$$\nu(a) = \frac{1}{1 + e^{-a}}. \tag{2.3}$$

The variables in the second layer z_j, $j = 1, \ldots, M$ are called hidden units or, collectively, as a hidden layer. Figure 2.1 shows a network diagram for a two-layer neural network with M neurons per hidden layer, D input variables and K output variables [12].

During the learning phase, the weights and biases of the neural network are estimated (learned) by minimizing the considered cost function J. The minimization of the cost function J is an iterative process, which makes it necessary to provide some stopping criteria. This is often done using a validation set of samples, where

the learning process stops when the squared error of prediction of the validation set stops decreasing. Overfitting is the phenomenon where further iterations lead to increasing squared error on the validation set, but decreasing error on the training set. One attempt to overcome overfitting is to use Bayesian regularized artificial neural networks [12, 13].

Bayesian regularized artificial neural networks

In Bayesian regularized artificial neural networks, the cost function J is defined as

$$J(\mathbf{w}) = \beta \sum_{n=1}^{N} ||y(\mathbf{x}_n, \mathbf{w}) - \mathrm{t}_n||_2^2 + \alpha \sum_{n=1}^{N} w_n^2, \tag{2.4}$$

with hyperparameters α and β. In (2.4), $||\cdot||_2^2$ denote the squared Euclidean norm, and $\{\mathbf{x}_n\}$ is the set of input vectors and $\{t_n\}$ denotes the target values, i.e., the true values, and N is the number of training data points in the learning set $\mathcal{L}$. Assume that the conditional distribution $p(t|\mathbf{x})$ for one target is Gaussian with $\mathbf{x}$-dependent mean given by output of the neural network $y(\mathbf{x}, \mathbf{w})$ with precision β, and that the prior distribution over the weights, $\mathbf{w}$ are

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}). \tag{2.5}$$

Furthermore, assume that the prior in (2.5) is Gaussian with zero mean and a precision of α [12, 13]. Let $\mathcal{D} = \{t_1, \dots t_N\}$, then

$$p(\mathbf{w}|\mathcal{D}, \alpha, \beta) \propto p(\mathbf{w}|\alpha) \prod_{n=1}^{N} \mathcal{N}(t_n|y(\mathbf{x}_n, \mathbf{w}), \beta^{-1}) \tag{2.6}$$

is the resulting posterior distribution is non-Gaussian, due to the non-linearity of $y(\mathbf{x}_n, \mathbf{w})$. In (2.6) $\mathcal{N}(\mu, \sigma^2)$ denotes a normal distribution with mean μ and variance σ^2 and the precision is $1/\sigma^2$. It is possible to find a Gaussian approximation of (2.6) using Laplace approximation with the local maxima found by numerical optimization. For example, use *Levenberg–Marquardt algorithm* to solve non-linear least-squares problems, and backpropagation to efficiently calculate the derivatives. The optimization on the logarithm of the posterior boils down to a least-squares problem.

Assuming the hyperparameters α and β are known it is possible to find a maximum a posteriori (MAP) estimator denoted by $\mathbf{w}_{MAP}$. From calculations done in [12], $p(\mathbf{w})$ is shown to have linear-Gaussian model with a Gaussian distribution and $p(t|\mathbf{w})$ is Gaussian distributed. Using the result of marginal and conditional Gaussians distributions in [12], the probability of a target given the input, target set, and hyperparameters are

$$p(t|\mathbf{x}, \mathcal{D}, \alpha, \beta) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}_{MAP}), \sigma^2(\mathbf{x})), \tag{2.7}$$

where

$$\begin{aligned} \mathbf{g} &= \nabla_{\mathbf{w}} y(\mathbf{x}, \mathbf{w})|_{\mathbf{w}=\mathbf{w}_{MAP}} \\ \sigma^2(\mathbf{x}) &= \beta^{-1} + \mathbf{g}^T(\alpha\mathbf{I} + \beta\mathbf{H})^{-1}\mathbf{g} \end{aligned} \tag{2.8}$$

with **H** denoting the Hessian matrix compromising the second derivatives of the sum-of-squared error comprising of **w**, and **I** is the identity matrix of appropriate size [12].

Hyperparameter optimization

So far we have assumed that the hyperparameters α and β are known and fixed. This is not always the case and they can be calculated by maximizing

$$\ln p(\mathcal{D}|\alpha,\beta) \simeq -J(\mathbf{w}_{MAP}) + \frac{1}{2}\ln|(\alpha\mathbf{I}+\beta\mathbf{H})| + \frac{W}{2}\ln\alpha + \frac{N}{2}\ln\beta - \frac{N}{2}\ln 2\pi. \quad (2.9)$$

with respect to α and β, where W is the number of parameters in **w**. Maximization of (2.9) is obtained using similar assumptions as for $\mathbf{w}_{MAP}$, for details see [12].

This leads to the values of hyperparameters as given as

$$\begin{aligned} \gamma &= \sum_{i=1}^{W} \frac{\lambda_i}{\alpha+\lambda_i} \\ \alpha &= \frac{\gamma}{\mathbf{w}_{MAP}^T \mathbf{w}_{MAP}} \\ \frac{1}{\beta} &= \frac{1}{N-\gamma}\sum_{n=1}^{N}\{y(\mathbf{x}_n,\mathbf{w}_{MAP}) - t_n\}^2, \end{aligned} \quad (2.10)$$

with λ_i is the ith eigenvalue of $\beta\mathbf{H}$. To find $\mathbf{w}_{MAP}$, both α and β have to be known and vice versa. This leads to the fact that the optimization is done by recursively updating the posterior distribution and re-estimating the hyperparameters [12, 13].

2.1.2 Random forest

Tree-based models are commonly used for classification and regression. The idea is to do binary decisions of the features in the input set to split it corresponding to different classifications. These models are often referred to as classification and regression trees (CART). An example of a single CART is shown in Figure 2.2. A nice property with such tree-based models as CART is that they are easy to illustrate and interpret [12].

The directed tree in Figure 2.2 takes two or three binary decision to split the input space {BRSRP, DoD } to estimate the output $\{pos_x, pos_y\}$. It will start from the root node (red) and works its way through the tree until it finds a leaf (white nodes). The depth of a tree is defined as the number of nodes passed in the longest path in the tree; for the tree in Figure 2.2 the depth is four.

It is very popular to generate many classifiers and aggregate the result over them. This method is called ensemble learning. Two well-known examples are *boosting* and *bagging*. In boosting, these classifiers get associated with a weight which will be updated through an iterative process. To understand bagging, or bootstrap

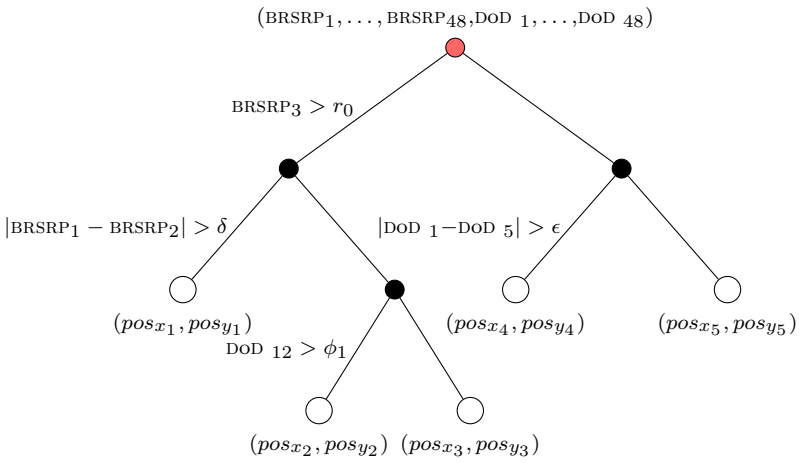


Figure 2.2: *Network diagram over a binary decision tree with a depth of four.*

boosting, let us first define a bootstrap sample.

A bootstrap sample is to create a set $\mathbf{X}_B$ by drawing N random samples from the data set $\mathbf{X} = \{\mathbf{x}_1 \dots \mathbf{x}_N\}$. This might lead to that some points in $\mathbf{X}$ are replicated in $\mathbf{X}_B$ while some are absent. Figure 2.3 is an example of taking three bootstrap samples from a limited learning set.

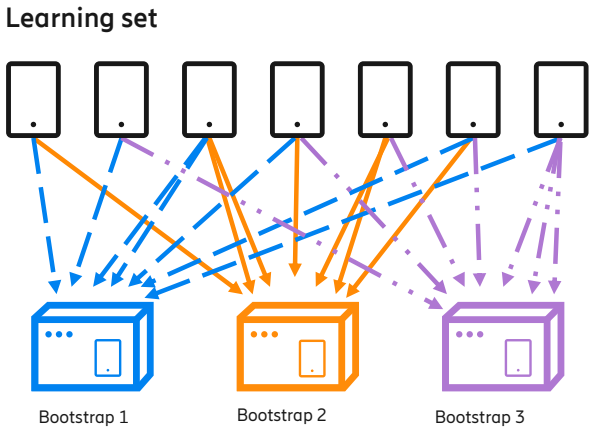


Figure 2.3: *Example of taking three bootstraps samples from a learning set consisting of seven data points. These data points are represented by mobile phones and the bootstraps samples with a boxes.*

Define bagging as taking repeated bootstrap samples from the learning set $\mathcal{L}$, construct a classifier in each of those sets, and take the final classifier as the average of all these smaller classifiers. Using the toy example in Figure 2.3, based on each box one classification tree is trained, and the bagging tree would be an average of those classifiers [12, 15, 16].

Random Forest adds an additional layer of randomness to bagging. In addition to the randomness in taking bootstrap samples, the random forest also changes the tree structure between iterations. For every kth tree, a vector Θ_k is generated, which has the same distribution as the past random vectors $\Theta_1, \ldots, \Theta_{k-1}$ but independent of them, this redom vector decides how the tree will split, hence, decide the structure of the tree [17]. Now define a random forest as a classifier that consists of a collection of tree-based modelled classifier $h(\mathbf{x}, \Theta_k), k = 1, \ldots, K_{tree}$, where Θ_k are independent identically distributed random vectors and each of the K_{tree} trees casts a unit vote for the most popular class at input $\mathbf{x}$ [16, 17]. The convergence of random forest is proven using the *strong law of large numbers* in [17]. Due to this, overfitting is not a problem for random forest when more trees are added to the forest.

Two extra pieces of information are also provided by random forest: variable importance and proximity measure. Variable importance measures how much the prediction error increases when the out-of-bag (OOB) data for that variable is changed while the others are left unchanged. An OOB classifier, is defined as classifiers whose learning set $\mathcal{L}_k \subseteq \mathcal{L}$ does not contain $\{t, \mathbf{x}\} \in \mathcal{L}$. In the proximity matrix, the (i, j) element tells us how often the ith and jth elements terminate to the same leaf. This defines the proximity measure and can give good insights into the structure of the data [16, 17].

In addition, a random forest is user-friendly with only two adjustable parameters (number of trees in the forest and the number of variables in the random subset at each node (depth of the tree)), it is also a very effective estimation tool. A random forest is robust to noise, faster than bagging and boosting, gives useful extra information such as variable importance, and it is easy to parallelize [12, 16, 17].

2.2 Detection of NLOS

To improve the performance of positioning in urban canyons, detection of NLOS conditions is central. Reliable detection opens up the possibility to use different positioning algorithms for LOS and NLOS. Less computational expensive LOS algorithms can be used in LOS conditions and a more computational heavier data-driven approach in NLOS conditions. Since LOS and NLOS have such different features, training NLOS positioning algorithms on LOS data will decrease the performance of the algorithm. Two different statistical detection approaches will be introduced in this section: Neyman-Pearson (NP) detector and generalized likelihood ratio test (GLRT).

2.2.1 Neyman-Pearson detector

The probability of detection (P_D) or power of the test, and probability of false alarm (P_{FA}) or level of significance or size of the test, are two important concepts in detection of a signal. Given the null hypothesis $\mathcal{H}_0$, the alternative hypothesis $\mathcal{H}_1$, and set of observations $\mathbf{x} = \{x[0], \ldots, x[N-1]\}$, the P_D and P_{FA} levels are given by

$$\begin{aligned} P_D &= \int_{R_1} p(\mathbf{x}; \mathcal{H}_1) d\mathbf{x} \\ P_{FA} &= \int_{R_1} p(\mathbf{x}; \mathcal{H}_0) d\mathbf{x}, \end{aligned} \tag{2.11}$$

where R_1 are the values that maps into deciding $\mathcal{H}_1$, or equivalently reject $\mathcal{H}_0$. The conditional PDF for the vector $\mathbf{x}$ under hypothesis $\mathcal{H}_1$ and the conditional PDF for the vector $\mathbf{x}$ under hypothesis $\mathcal{H}_0$ are denoted $p(\mathbf{x}; \mathcal{H}_1)$ and $p(\mathbf{x}; \mathcal{H}_0)$, respectively. Neyman-Pearson's theorem states that to maximize P_D for a given P_{FA}, then decide on hypothesis $\mathcal{H}_1$ if

$$L(\mathbf{x}) = \frac{p(\mathbf{x}; \mathcal{H}_1)}{p(\mathbf{x}; \mathcal{H}_0)} > \gamma \tag{2.12}$$

where the threshold γ is found from

$$P_{FA} = \int_{\{\mathbf{x}: L(\mathbf{x}) > \gamma\}} p(\mathbf{x}; \mathcal{H}_0) d\mathbf{x}. \tag{2.13}$$

This is referred to as the NP detector [18]. Hence, there is a trade off between desired P_D (high) and P_{FA} (low). In Figure 2.4 this is illustrated using the PDF of a signal $\mathbf{x}$ that is Gaussian both under $\mathcal{H}_1$ and $\mathcal{H}_0$.

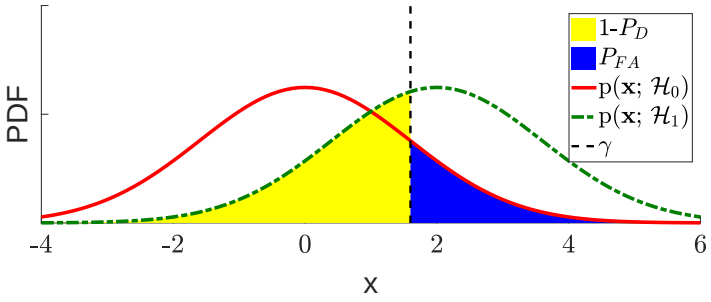


Figure 2.4: *Probabitity of Detection and False Alarm.*

As seen in Figure 2.4, a increase of P_D will lead to an increase in P_{FA} and vice versa. This since if the threshold γ would move to the right the green area would be smaller, but so would the complement to the yellow area under the red curve.

Detection set-up

In this thesis, the problem of detecting NLOS is modelled as detection of direct current (DC) level in white Gaussian noise (WGN). The DC level here correspond to a decay in BRSRP, or a bigger difference in DOD between the antenna elements with strongest BRSRP. That means $x[n]$ is the difference in ether BRSRP or DOD between the strongest beams, see Section 3.4. The null hypothesis, $\mathcal{H}_0$, is that the UE is in LOS and the alternative hypothesis, $\mathcal{H}_1$, is that the UE is in NLOS.

$$\begin{aligned} \mathcal{H}_0 &: x[n] = w[n] \ , \ n = 1, \dots, N-1 \\ \mathcal{H}_1 &: x[n] = A + w[n] \ , \ n = 1, \dots, N-1 \end{aligned} \tag{2.14}$$

In (2.14), where the amplitude when entering NLOS conditions $A > 0$ is known and $w[n]$ is WGN with a known variance σ^2. Under these circumstances $\mathbf{x} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ under $\mathcal{H}_0$ and $\mathbf{x} \sim \mathcal{N}(A, \sigma^2\mathbf{I})$ under $\mathcal{H}_1$, where $\mathcal{N}$ is the normal distribution and $\mathbf{I}$ the identity matrix [18].

This leads to the set-up equivalent to the detection of a change in the mean of a multivariate Gaussian PDF. Using the NP detector we get the detection rule: that we should decide $\mathcal{H}_1$ if

$$\frac{\frac{1}{(2\pi\sigma)^{\frac{N}{2}}} e^{-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}(x[n]-A)^2}}{\frac{1}{(2\pi\sigma)^{\frac{N}{2}}} e^{-\frac{1}{2\sigma^2}\sum_{n=0}^{N-1}x^2[n]}} > \gamma, \tag{2.15}$$

which can be simplified to

$$\frac{1}{N}\sum_{n=0}^{N-1} x[n] > \frac{\sigma^2}{NA}\ln\gamma + \frac{A}{2} = \gamma'. \tag{2.16}$$

From this also a relationship between P_{FA}, P_D, and the threshold γ' can be deduced using the complementary cumulative distribution function $Q(x)$ defined as

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}t^2} dt. \tag{2.17}$$

For a given P_{FA}, the following relationships can be used to calculate P_D and γ':

$$\begin{aligned} \gamma' &= \sqrt{\frac{\sigma^2}{N}} Q^{-1}(P_{FA}) \\ P_D &= Q\Big(Q^{-1}(P_{FA}) - \frac{NA^2}{\sigma^2}\Big). \end{aligned} \tag{2.18}$$

2.2.2 Generalized likelihood ratio test

Consider the same problem formulated as in (2.14) but that the parameter A is unknown, in which case the NP detector cannot be used. Instead of using A,

replace the unknown parameter with its maximum likelihood estimate (MLE); this approach is called generalized likelihood ratio test (GLRT). Let $\hat{\theta}_i$ denote the MLE of the unknown parameter θ_i under hypothesis $\mathcal{H}_i$. Then, the GLRT decides $\mathcal{H}_1$ according if [18]

$$L_G(\mathbf{x}) = \frac{p(\mathbf{x}; \hat{\theta}_1, \mathcal{H}_1)}{p(\mathbf{x}; \hat{\theta}_0, \mathcal{H}_0)} > \gamma. \tag{2.19}$$

Detection set-up

Consider the DC level in WGN with unknown amplitude $\theta_1 = A$ and $\theta_0 = 0$, then the hypothesis becomes

$$\begin{aligned} \mathcal{H}_0 &: A = 0 \\ \mathcal{H}_1 &: A \neq 0. \end{aligned} \tag{2.20}$$

In [14], it is shown that the MLE of A is equal to

$$\hat{A} = \bar{x} = \frac{1}{N} \sum_{n=0}^{N-1} x[n], \tag{2.21}$$

and the detection becomes: decide $\mathcal{H}_1$ if

$$\bar{x}^2 > \frac{2\sigma^2 \ln \gamma}{N} = \frac{\gamma'}{N^2}. \tag{2.22}$$

Denote $Pr(\,\cdot\,)$ as the probability of an event. Using the $Q(x)$ function, (2.17), and that

$$P_{FA} = Pr\big(|N\bar{x}| > \sqrt{\gamma'}; \mathcal{H}_0\big), \tag{2.23}$$

γ' and P_D can be calculated for a given P_{FA} as

$$\begin{aligned} \sqrt{\gamma'} &= \sqrt{\sigma^2 N} Q^{-1}\Big(\frac{P_{FA}}{2}\Big) \\ P_D &= Q\Big(Q^{-1}(P_{FA}/2) - \sqrt{N\hat{A}^2/\sigma^2}\Big) + Q\Big(Q^{-1}(P_{FA}/2) + \sqrt{N\hat{A}^2/\sigma^2}\Big). \end{aligned} \tag{2.24}$$

2.3 Kalman Filter

To improve the accuracy of the positioning, filtering the output from the machine learning algorithms using a Kalman filter has been investigated. Kalman filter is used for solving the prediction, filtering and smoothing problem. This is done in such a way that the computation time is low, so that it is possible to implement a Kalman filter that runs in real time [19]. In this thesis the filtering aspect of the Kalman filter will be used. Assuming that process noise and measurement noise are Gaussian, Kalman filter is the best possible estimator among both linear and non-linear estimators [19]. This assumption is considered in this thesis.

3

Methods

In this chapter, the methods and different algorithms are explained in a way they have been applied in this study. The complete problem set-up is also presented.

3.1 Data Description

This section describes the testbed data used in this thesis, the scenario during which the data was collected, and how the data was collected. Processing of data and selection of features to use in the machine learning algorithms and for detection, will also be discussed. Since one big limitation in this work is the relatively small size of the data sets, techniques to generate new set of data will also be described.

3.1.1 Scenario

The scenario used in this thesis is a UE moving around in an urban area in Kista, Stockholm. The UE is a car equipped with an antenna which communicates with the TP. It is moving at walking speed (around 7 km/h). Illustration of the scenario and a map of the area of positioning has been previously given in Figure 1.1 and Figure 1.2b, respectively. In Figure 3.1, a picture of the UE and TP used in the testbed are shown. The carrier frequency of the TP is 15 GHz and the antenna of the TP used in the testbed consists of two 8×8 antenna grid. Part of the antenna grids are used in the digital beamforming to create 48 beams, with horizontal beam widths of 6°and vertical beam widths of 5°. Beams are placed into five different vertical layers with nine to ten beams in each layer. The beam-grid is shown in Figure 3.2.

(a) UE *(b)* TP

Figure 3.1: ***(a)****Picture of the* UE *used in the testbed.* ***(b)*** *Picture of the* TP *used in the testbed.*

In the logged data set, the UE has travelled along three different paths. In total there are around 1200 data points including LOS and NLOS conditions. The part with only NLOS is around 400 data points. The part of data where multiple samples for one position is available is very limited.

The logged data consists of the BRSRP for all 48 beams, sampled at 10 ms intervals, and position data logged by a global positioning system (GPS) receiver at sample rate of 10 Hz. Hence, the distance between consecutive measurements is around 0.19 m. To get the same length of the input and target, the BRSRP data is

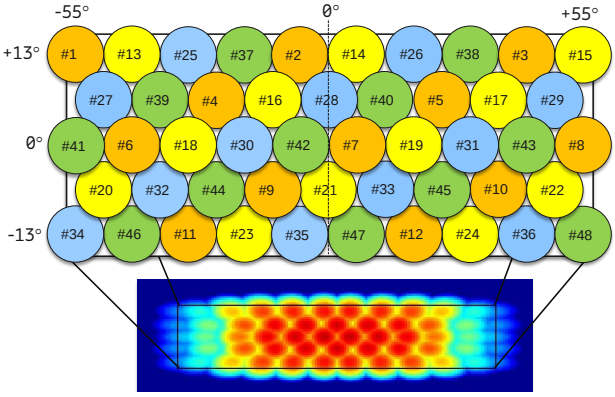


Figure 3.2: *Beam-grid of the* TP *with carrier frequency 15 GHz. In the lower part of the picture, the more red the colour are the higher is the power of the signal.*

averaged over ten samples. This can be considered as low-pass filtering the input. Worth noticing is that the error of the GPS position for a UE under open sky can be up to 5 meters, and even larger in urban canyons [20].

From the logged GPS positions given in longitude and latitude the distance from the UE to the TP is calculated. Then the coordinate system is changed from a longitude/latitude-coordinate system to a Cartesian coordinate system with TP position in origin. This will from now on be the referred coordinate system used in this thesis.

3.1.2 Selection of the best beam

For every time instant, the n beams with highest BRSRP are chosen. The number of beams n that has been investigated in this thesis are chosen between five and fifteen. This due to the fact that information from beams with low BRSRP are unreliable, and they add complexity to the algorithms while provide marginal performance improvement. This is also consequence of the interpolated constraints and time limitation for this thesis. An example of the beam selection process is shown in Figure 3.3.

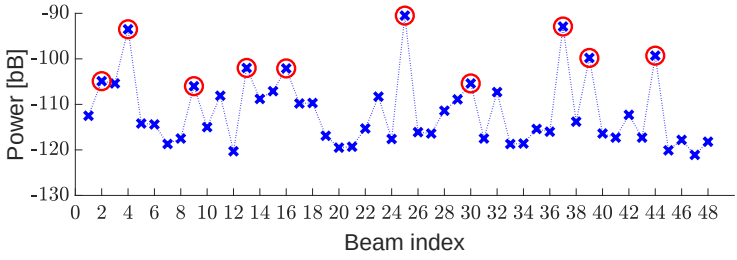


Figure 3.3: *Selection of the beams with highest signal power. Here the 10 beams with highest BRSRP are chosen.*

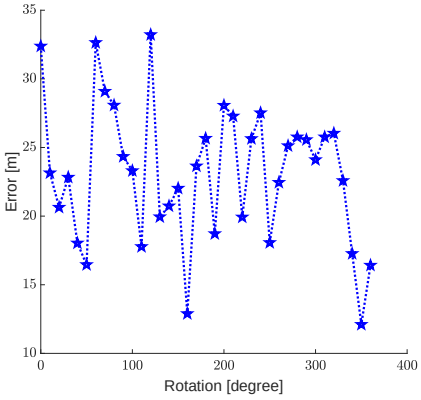
3.1.3 Rotation of reference frame

During the test of positioning using neural networks a big difference in performance between x-and y-direction was discovered. To minimize the positioning error

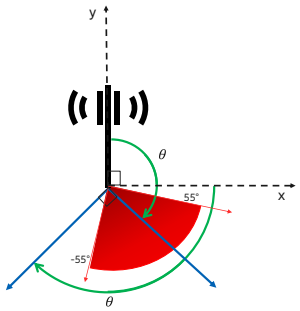
$$\sqrt{(\widehat{pos}_x - pos_x^{true})^2 + (\widehat{pos}_y - pos_y^{true})^2}, \tag{3.1}$$

rotation of the reference frame has been investigated, where $\widehat{pos}$ is the estimated position. Results are shown in Figure 3.4a.

From Figure 3.4a the rotation of the frame of reference is chosen as 160°. The rotation of 160° coincides with the difference angle between the reference frame for the UEs path as a function of the distance from the TP and the path given in



(a) Performance for different angel of rotation. On the $y-$axis is the absolute error of position and on the $x-$axis the angle of rotation.



(b) Picture describing rotation of reference frame. The red arc is the range in which the TP has antennas with coverage, -55°to 55°.

Figure 3.4: *Rotation of reference frame.*

GPS coordinates. In Figure 3.4b, a schematic picture of the rotation is shown. The coverage of the antenna is marked with red in Figure 3.4b and the old coordinate system with black doted axis and after rotation with blue axis.

3.1.4 Feature selection

From the BRSRP many different features can be calculated. Evaluation of feature importance is done with the help of random forest algorithms, described in Section 2.1.2. Only snapshot features are used as input features to the algorithms. In Table 3.1 thefeatures used in this thesis are listed.

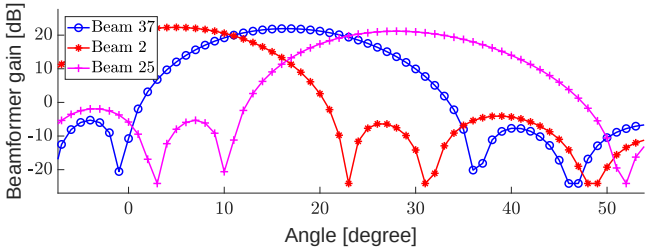
Table 3.1: *Features classes that are selected as inputs to the machine learning algorithms.*

Feature	Description
BRSRP	Beam reference signal received power, which is defined in [21].
DoD	Direction of departure.
DBRSRP	Difference in BRSRP between strongest and consecutive strongest beams.
DDoD	Difference in DoD between strongest and consecutive strongest beams.

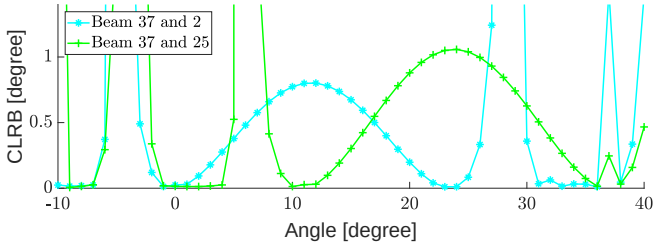
Estimation of direction of departure

To estimated direction of departure (DoD), i.e., the angle of the beam on the antenna which the signal departures from, the beam pattern of the antenna has to be calculated. This calculation is done using similar algorithms as in [2], with

BRSRP as input to the algorithm. The estimated DoD for three different beams are shown in Figure 3.5a and Cramér-Rao Lower Bound (CRLB) that estimator selects the correct angle given two alternative is calculated using results from [5, 9], and shown in Figure 3.5b.



(a) *Beamformer gain as a function of angle for three adjacent beam patterns. Estimation of DoD is the argmax of the beamformer gain.*



(b) *CRLB for DoD between beam 37 beam and two of its beams that are adjacent to it.*

Figure 3.5: *Estimation of DoD and CRLB between estimation for one beam and its neighbouring beams.*

Where on the antenna the beams whose DoD are estimate in Figure 3.5a can be seen in Figure 3.2. The beams selected such that they are next to each other in the antenna, to demonstrate the resolution of DoD. More details on the algorithm used for estimation of DoD are found in [3, 4]. For calculations of the CRLB in Figure 3.5b, methods from [5, 9] are used. There, it is proven that the CRLB of selecting the right angel for two beams as candidates can be calculated as

$$\mathrm{Var}(\phi) = \frac{2\sigma^2_{\mathrm{BRSRP}}}{\left(\frac{dH^{ij}(\phi)}{d\phi}\right)^2}, \tag{3.2}$$

where ϕ is the estimation of DoD, σ^2_{BRSRP} is the variance of the measurement noise, and $H^{ij}(\phi)$ is the difference in beamformer gain between beam i and j. Here, variance of the measurement noise is choosen as 1 dB for implementation purposes. In Figure 3.5b one can see that when the estimation gets to a side lobe the CRLB becomes very large.

3.1.5 Generation of larger set of data

As described in Subsection 3.1.1 the amount of data is very small. Therefore, ways to generate new sets of data has been investigated. Two different ways to generate new sets of data have been investigated and tested. These are interpolation based on the difference in sampling rate, and using the fact that the antenna has different vertical layers.

Interpolation

Due to the difference in sampling rate between the BRSRP data and the GPS data, it is possible to get more data by interpolating the GPS receiver data. The sampling period of the BRSRP and the GPS measurements are 10 ms and 100 ms, respectively. This factor ten between the sampling rates makes it possible to interpolate and create ten times more data. In this case, ten consecutive BRSRP get the same target value.

Separation of vertical beam layers

The horizontal position of the UE is of interest in this thesis, and the vertical position will not be considered. Since the antenna have five different vertical layers of beams, see Figure 3.2, one idea to is to generate more data by assuming that every vertical layer of beams are independent data sets. This method will also increase the amount of data with a factor of five.

3.1.6 Performance metric

The performance metric used here is based on the one presented in the indoor positioning study item in 3rd Generation Partnership Project (3GPP) [22]. The performance metric mentioned in that 3GPP report are the values at which the cumulative distribution function (CDF) of the positioning error reach 40%, 50%, 70%, 80% and 90%. In this report, the focus will be on 80%.

Since information about the probability density function (PDF) of positioning error is missing, the CDF can not be calculated. Therefore, instead of using the CDF as performance metric, its unbiased estimator empirical cumulative distribution function (ECDF) is used. The definition of the ECDF is

$$\widehat{F}_n(t) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{X_i \leq t}, \tag{3.3}$$

where $X_1, \cdots X_n$ are independent, identically distributed random variables with the common PDF $f_X(y)$, and $\mathbf{1}_{X_i \leq t}$ is the indication function for the event $X_i \leq t$, e.g the function that is one when the event $X_i \leq t$ happens and zero otherwise. Then it can be shown using law of large numbers that $\hat{F}_n(t) \xrightarrow{a.c.} F(t)$, [23]. For the sake of notational simplicity, throughout the rest of the report no distinction will be made between the CDF and the ECDF.

3.2 Neural Networks

The neural networks are implemented in MATLAB using the neural network toolbox. The task is translated into a regression problem where the goal is to fit a function from the input features to the target values. The neural network used in this thesis have two hidden layer and an activation function $h(\cdot)$ that in the first layer is a $tanh()$, and a activation function in the second layer is pure line.

3.2.1 Size of hidden layers

To determine the size of the hidden layers to use in a neural network, networks with different number of neurons were implemented. Then the performance of all the networks on evaluation data were calculated and the number of neurons, or size of hidden layer, that gave the best performance (as defined in Section 3.1.6) for its complexity were chosen. The size of hidden layers are chosen according to Figure 3.6 and the values are summarized in Table 3.2.

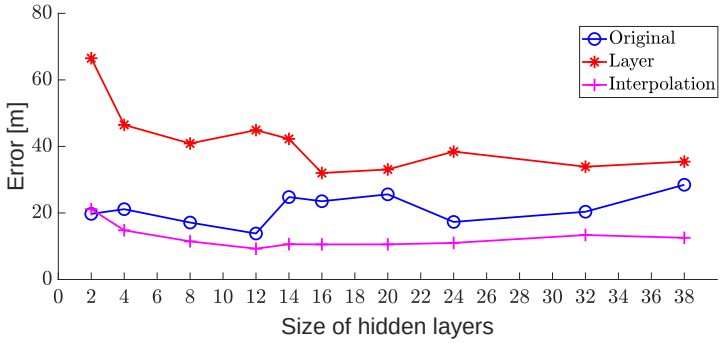


Figure 3.6: *Selection of number of neurons in the hidden layers for the different set of data.*

Table 3.2: *Number of neurons in the hidden layer for the different data set.*

Set of data containing:	**Number of neurons in the hidden layers**
Original data	12
Interpolation data	12
Separated vertical beam layer	14

3.2.2 Combining multiple networks

To prevent overtraining and improve the performance, a technique of combining result from multiple networks are used. Multiple neural networks with the same number of neurons and layers are given same set of learning data to train on.

After the learning, all networks are tested with the same test data and the output is averaged over all networks. Then the average results are validated against the true position of the UE. Define the average error as

$$\bar{e} = \sqrt{\Big(\frac{1}{M}\sum_{m=1}^{M}\widehat{pos}_x^m - pos_x^{true}\Big)^2 + \Big(\frac{1}{M}\sum_{m=1}^{M}\widehat{pos}_y^m - pos_y^{true}\Big)^2} \tag{3.4}$$

where M is the number of neural networks, $(\widehat{pos}_x^m, \widehat{pos}_y^m)$, $m = 1, \cdots, M$ are outputs from the individual networks, $(pos_x^{true}, pos_y^{true})$ is the true position, and $\bar{e}$ is the average error. There are many other examples of successfully combining multiple networks to improve performance of neural networks [24–26].

3.2.3 Pre-processing of features

To improve performance of neural networks some processing of inputs and targets are done. Techniques used in this thesis includes: normalizing data to be between $[-1, 1]$ with zero mean, and unity variance together with removing constant features. Description of various preprocessing techniques and its effect on machine learning algorithms can be found in [27]. After the regression is done, the output will be converted to the original units.

3.3 Random Forest

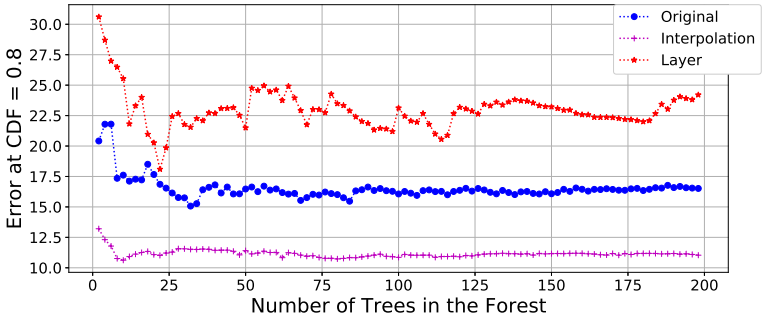
The random forest is implemented in PYTHON using *scikit learn*. The problem here (as well as in the case of neural networks) is a regression problem, and so the Random Forest Regression library in *scikit learn* is used.

3.3.1 Number of trees in the forest

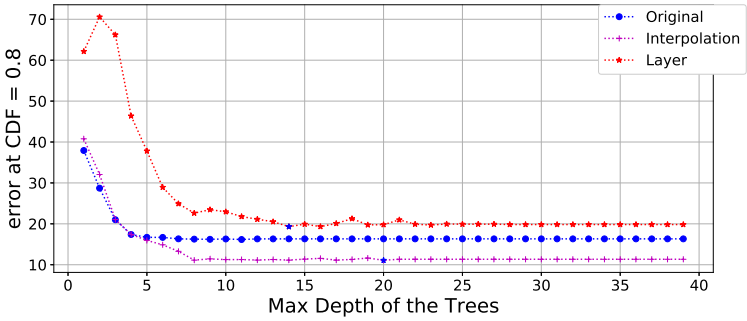
There is a threshold in number of trees used in a random forest, when increasing the number of trees does not lead to big improvement in performance. This is according to results found in [28]. To find this sweetspot in number of trees simulations were done. Result can be seen in Figure 3.7a. From this, one can see that after 125 trees the performance converges, hence the number of trees are chosen to 125 for all set of data.

3.3.2 Depth of the trees

Different depth of the trees have been investigated and the results are shown in Figure 3.7b. Worth mentioning is that variation of the depth does not effect the error of the positioning error significantly. Since the performance converges after a depth of ten trees for all set of data, the depth are chosen as ten.



(a) Results from simulation for how many trees is optimal in the forest. One can see that the performance have converge after the forest consist of 125 trees for all set of data.



(b) Results from simulation for the depth of the trees. One can see that the performance converges around 10 trees for all set of data.

Figure 3.7: *Selection of parameters for random forest.*

3.3.3 Pre-processing of features

Similar to the neural network, pre-processing of features were done to improve performance. Input features are centred around the mean and normalized to unit variance.

3.4 Detection of NLOS

In this section, method and implementation of the in Section 2.2 outlined methods for detecting NLOS conditions are described. What signals are used for detection, how the parameters A and σ^2 are obtained and approximated, and also how to compute the threshold γ and P_D for a given P_{FA}. The detection algorithms are implemented in MATLAB.

3.4.1 Signal selection

The two different signals available for detection of NLoS are BRSRP and DoD. The first signal used for detection, $x_1[n]$, is using the difference in BRSRP between beams, and the second signal, $x_2[n]$, are using the difference in DoD between beams. For the first case using BRSRP the signal is defined as

$$x_1[n] = \frac{1}{M-1} \sum_{m=2}^{M} \text{BRSRP}_1[n] - \text{BRSRP}_m[n],. \tag{3.5}$$

where $\text{BRSRP}_1[n]$ is the BRSRP of the strongest beam at time instance n, and $\text{BRSRP}_m[n]$ $m = 2, \ldots, M$ is the BRSRP of the M consecutive strongest beams. From this signal offsets is chosen such that the signal in LoS conditions is around zero. This so that when the UE enter a region with NLoS conditions there will be a step in the signal. The signal is also normalized to have an amplitude of one when entering NLoS. For the second case using DoD the signal is defined as

$$x_2[n] = \frac{1}{M-1} \sum_{m=2}^{M} |\text{DoD}_1[n] - \text{DoD}_m[n]|. \tag{3.6}$$

where $\text{DoD}_1[n]$ are the DoD of the strongest beam at time instance n, and $\text{DoD}_m[n]$ $m = 2, \ldots, M$ are the DoD of the M consecutive strongest beams. The signal is also normalized to have an amplitude of one when entering NLoS. For both signal the number of beams M is chosen as 9. This choice is based on investigation of the data, and when adding new beams did not provide new useful information.

3.4.2 Parameter selection

Using the NP detector the amplitude A is chosen by studying $x_1[n]$ and $x_2[n]$ and how they behave where NLoS conditions are expected. The variance σ^2 is chosen as the maximum variance of the signals, $x_1[n]$ and $x_2[n]$, after classification if they were in NLoS conditions for both the NP detector and the GLRT. In Table 3.3 all parameters used for detection of NLoS are listed.

Table 3.3: *Parameter selection for detection of NLoS.*

	BRSRP		DoD	
	NP	GLRT	NP	GLRT
A	0.5	-	0.5	-
σ^2	0.05	0.05	0.15	0.15
P_{FA}	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$	$3 \cdot 10^{-5}$
N	7	7	10	10

In Table 3.3, N refers to the number of beams per classification, which is choose so that the resolution for detection gets precise enough but not still the trend of the signal is captured. The probability of false alarm P_{FA} is consider a design

parameter and chosen small. From P_{FA}, the threshold γ' and probability of detection P_D are calculated according to (2.18) and (2.24). In Figure 3.8 the receiver operating characteristic for the two NP detectors are shown.

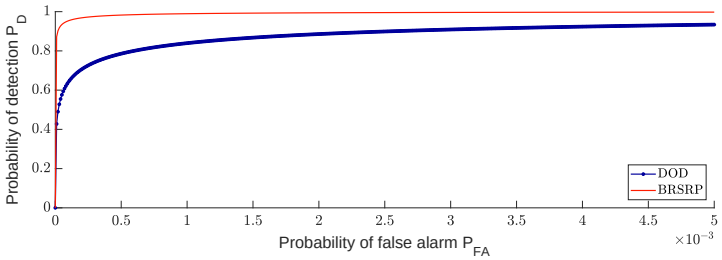


Figure 3.8: *Receiver operating characteristic for the model used in the NP detector.*

3.5 Kalman Filter

In this section, the set-up of the Kalman filter is described, and its initial parameters are specified. The choice of state-space model and the reason for selecting it are also elaborated. The Kalman filter is implemented according to algorithms found in [19].

3.5.1 State-space model

That the UE is moving at constant speed is previously mentioned in Subsection 3.1.1. This information is used when creating the state-space model. Using the relation for the distance for a moving object with constant velocity the system matrix and noise matrix are modelled according to [19, 29]. The sampling time, T_s, of the GPS position is 100 ms, see Subsection 3.1.1.

3.5.2 Set-up

As stated in Section 2.3, the process noise, and the measurements noise are assumed Gaussian, i.e., the estimation error of the machine learning algorithms are Gaussian distributed. Assume that the covariance matrix of the process noise is $0.01 \cdot \mathbf{I}_{2x2}m^2$ and that covariance matrix of the measurements noise is $25 \cdot \mathbf{I}_{2x2}m^2$. The choice of precess noise comes from that we trust the machine learning algorithms while we know that the GPS might have an error på to 5 m, hence measurements noise of 25 m^2. The initialization parameters are chosen as the first position with zero speed, $[pos_{x,1}, pos_{y,1}, 0, 0]$, and the initial covariance matrix chosen as $100 \cdot \mathbf{I}_{4x4}$, based on investigations of different values of the initial covariance matrix.

4

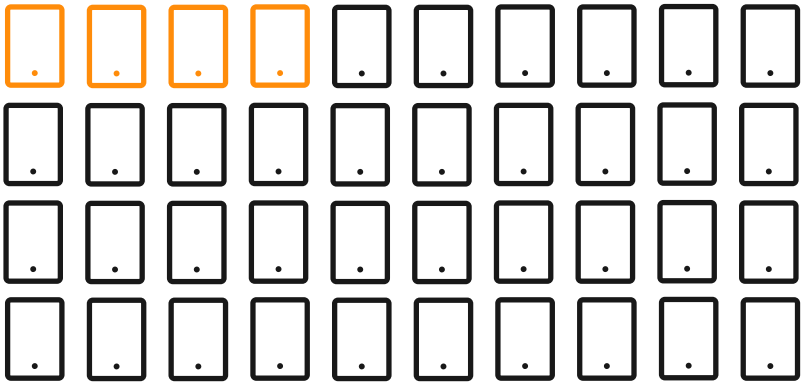
Performance Evaluation

This chapter presents the results of the evaluation of the machine learning algorithms. The feature vector used in these experiments consists of BRSRP for the n beams with highest BRSRP, defined as *best beams*, DoD of the n best beams, difference in BRSRP between the best beam and consecutive ones, and difference in DoD between the best beam and consecutive ones. For the data set where vertical beam layers in the antenna are separated, the number of best beams are set to five. For the original data, and the interpolated data, the number of best beams are set to ten. These numbers are selected such that performance does not improve significantly when more best beams are added.

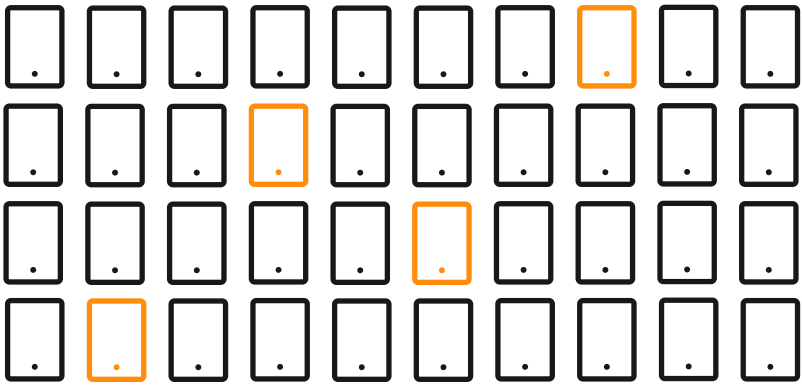
For the evaluation, around 10% of the data is used and the remaining 90% of the data is used for training. Performance of the machine learning algorithms are defined as the value when CDF of the positioning error is 0.8. In appendix B, a table summarising results from all machine learning algorithms is provided.

There are two different ways to separate the data set into a learning set and a testing set, these are consecutively or randomly, see Figure 4.1. For comparison, positioning in LoS conditions using machine learning algorithms designed similar to the one used for positioning NLoS are investigated.

Results from detection of NLoS conditions are presented, both using a NP detector as well as using a GLRT. This chapter also includes results from investigation of filtering the output from the machine learning algorithms to improve the positioning accuracy.



(a) Consecutively



(b) Randomly

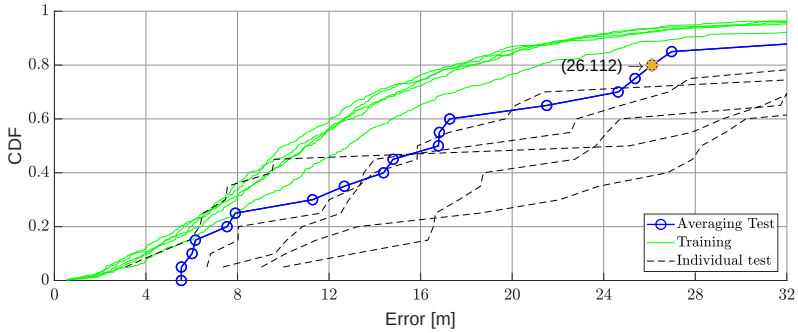
Figure 4.1: *Example of the two ways to split the dataset to evaluation and learning data - consecutively (a) and randomly (b). The orange phones are used as testing set.*

4.1 Neural Networks

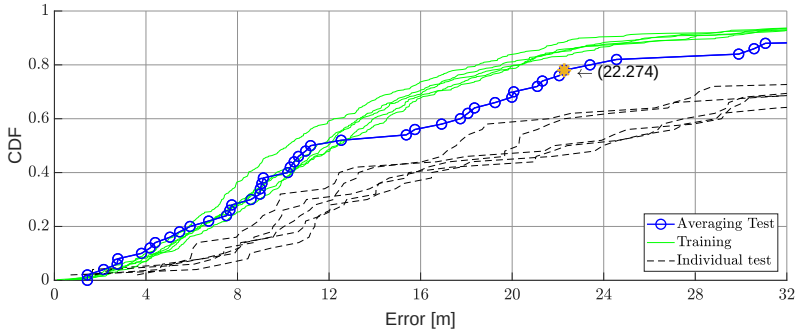
In this section the results obtained by neural networks are described. In Figure 4.2-4.4 the solid line is the performance combining output from over 100 neural networks, green is the performance on the learning data and black denotes the individual performances of the neural networks. Five of the individual tests and learning performances are selected for presentation. The number of neurons in the hidden layers are chosen according to Section 3.2.1 and description of design of the neural network can be found in Section 3.2.

4.1.1 Original data

Results for neural networks with learning set consisting of original data set is shown in Figure 4.2. The performance evaluated with consecutively and randomly seperation of learning and testing set are shown in Figure 4.2a and Figure 4.2b, respectively.



(a) *Performance evaluated on test data separated consecutively from the data set.*



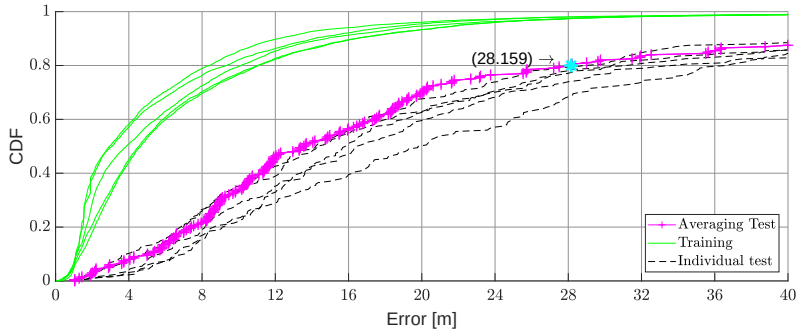
(b) *Performance evaluated on test data separated randomly from the data set.*

Figure 4.2: *CDF of the positioning error for neural networks trained and evaluated with original data set. One can see that there is only a few meters difference between the performance of two ways, consecutively and randomly, to separate the data set in learning set and evaluation set.*

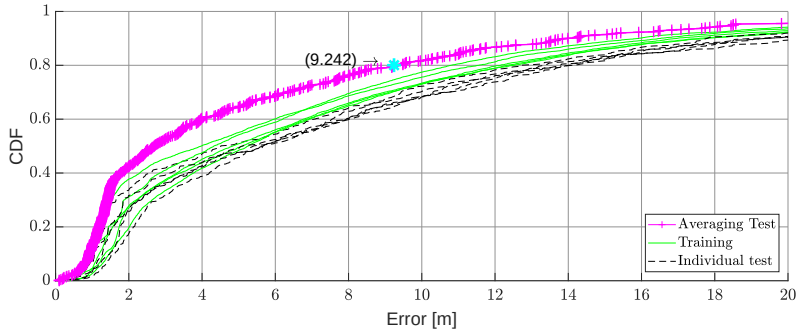
As one can see, the performance of averaging results over multiple neural networks boosts the positioning performance on evaluation data. It is also possible to see that performances are similar for testing set separated consecutively and randomly from the original data set.

4.1.2 Interpolation of data

Results for neural networks with data set consisting of interpolation data are shown in Figure 4.3. The performance evaluated with consecutive and random separation of learning and testing set are shown in Figure 4.3a and Figure 4.3b, respectively.



(a) *Performance evaluated on test data separated consecutively from the data set.*



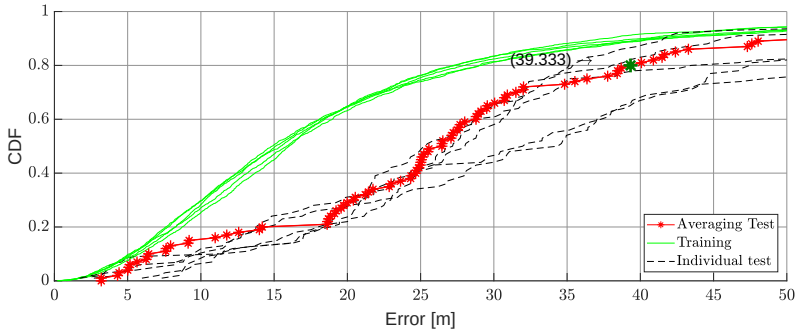
(b) *Performance evaluated on test data separated randomly from the data set.*

Figure 4.3: *CDF of the positioning error of neural networks trained and evaluated on data set created utilising the difference in sampling rate between BRSRP and GPS. Due to the better performance for test set separated randomly, the scale on the x-axis the double the size in Figure 4.3a compared with Figure 4.3b.*

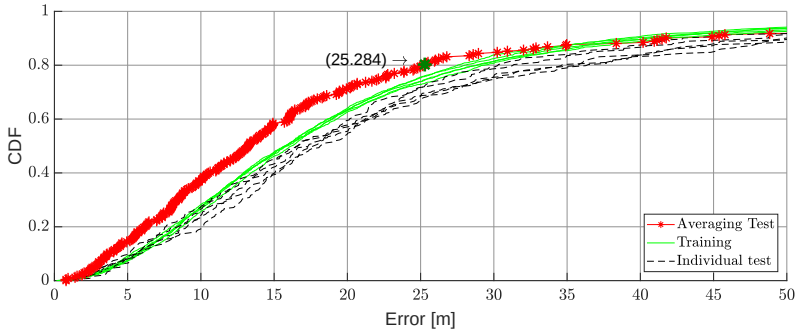
In Figure 4.3a there is a strange behaviour of the CDF. Results are very poor compared to Figure 4.3b, and the CDF is almost a straight line after a certain number of estimated positions. This might be a consequence of the small learning data set. Since the learning set is small, the learning set might miss points similar to the target values given in the test data set. This results in that all estimations might end up at the same point, far away from the true value. In Figure 4.3b the use of multiple networks boost the performance, see Section 3.2.2.

4.1.3 Separation of vertical beam layers

Results for neural networks with data set in which different vertical layers are separated are shown in Figure 4.4. The performance evaluated with consecutively and randomly separation of learning and testing set are shown in Figure 4.4a and Figure 4.4b, respectively.



(a) Performance evaluated on test data separated consecutively from the data set.



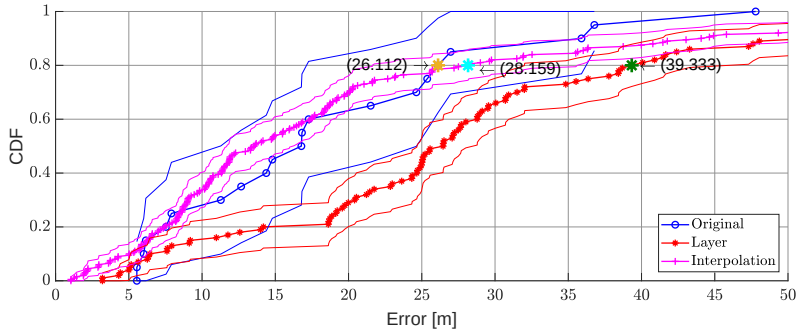
(b) Performance evaluated on test data separated randomly from the data set.

Figure 4.4: *CDF of positioning error for neural networks trained and evaluated with data set where vertical beam layers in the TP are separated. One can notice that there is more then ten meters better performance when learning set and testing set are separated randomly.*

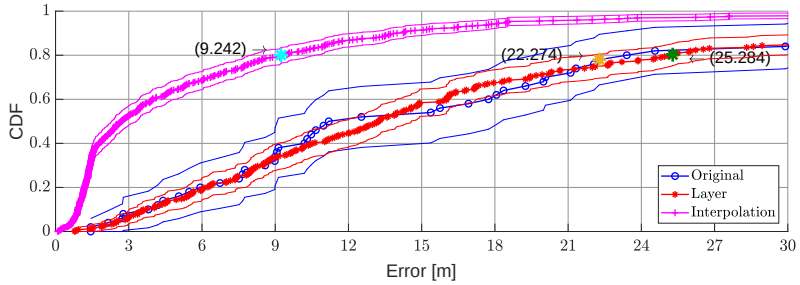
As expected combining output from multiple neural networks boosts the performance, see Subsection 3.2.2. Please note as well that the performance using beams from only one layer is much worse than evaluation on the original data set, see Figure 4.2 and Figure 4.4, where information from multiple layers are used. This would indicate that using information from multiple layers are essential for positioning with high accuracy.

4.1.4 Comparison of learning sets

In Figure 4.5 performance for different learning sets are shown; both for randomly and consecutively separation the learning and testing data. The plots are the same as the blue line in Figure 4.2, magenta in Figure 4.3, and red in Figure 4.4, that is the positioning accuracy after combining output from multiple neural networks.



(a) *Performance evaluated on test data separated consecutively from the data set.*



(b) *Performance evaluated on test data separated randomly from the data set.*

Figure 4.5: *Comparison of the performance for neural networks trained and evaluated different data sets. Since the positioning accuracy is much better for the randomly separation of learning and testing set, the scale on the x-axis for the randomly separated is smaller.*

A 95% confidence interval for the CDFs calculated using *Greenwood's Formula* are shown for all the data set as well. The confidence is represented by two lines, in the same colour as the data which the data set is presented. Confidence interval for the original data set is significant larger than for the others; this might be due to the limited number of data points in the original data set. The point where the CDFs reach 0.8 are highlighted in Figure 4.5a and Figure 4.5b.

4.2 Random Forest

This section presents the positioning result of using the random forest algorithm. Both the performance for different input features and the ranking of their importance are provided.

4.2.1 Feature importance

By using the random forest algorithm, it is possible to get a ranking of the features. That is how important the features are to get the predicted value. Features tested are BRSRP, DoD, difference between the beam with highest BRSRP and the consecutive ones (DBRSRP), and difference in DoD between the best beam and the consecutive ones (DDoD) for all n chosen best beams. Results are shown in Table 4.1 and Table C.1 and Table C.2 in appendix C, where higher score indicates that the feature is more important.

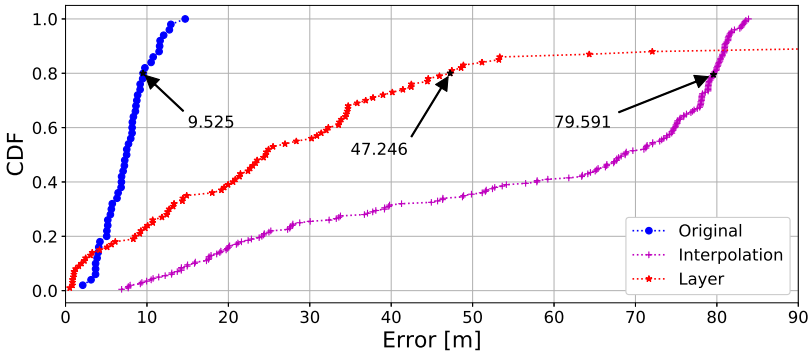
Table 4.1: *The ranking of feature importance generated by the random forest algorithm for the original data set. The best beam is the beam with highest BRSRP. The feature with highest importance is highlighted.*

Best beam nr.	BRSRP	DoD	DDoD	DBRSRP
1	0.0034	0.0102	-	-
2	0.0049	0.0161	0.0220	0.0032
3	0.0043	0.0052	0.0026	0.0218
4	0.0037	0.0251	0.0028	0.0348
5	0.0062	0.0652	0.0073	0.0041
6	0.0022	0.0071	0.0093	0.0031
7	0.0049	0.0086	0.0040	0.0118
8	0.0106	0.0016	0.0058	0.0186
9	0.0170	0.0019	0.0058	0.0128
10	0.0069	0.0040	0.0057	**0.6196**

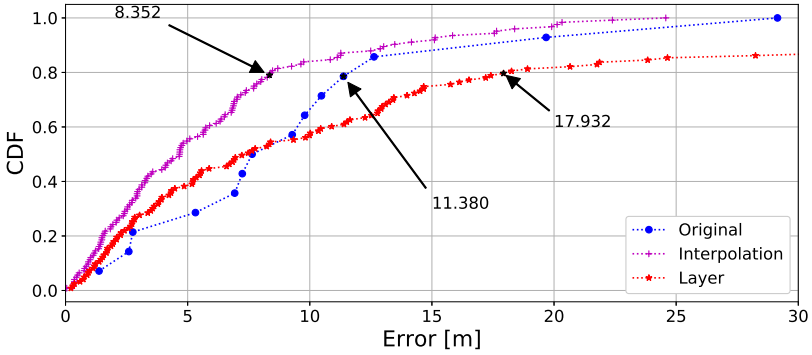
Worth noticing that for all data sets the last feature, that is highlighted one, has a very high importance. This opens up for investigation if data from more beams would improve performance.

4.2.2 Comparison of learning sets

Figure 4.6 shows result from performance of random forest algorithm evaluated on different data sets. These forests are both trained and evaluated on learning sets and testing sets that are separated consecutively and randomly. Worth noticing is the poor performance of the interpolation on consecutively separation of learning and testing data compared to randomly separated. This might be due to a lack of data, or lack of training data on a specific area along the path the UE are travelling. This statement is supported by the fact that positioning error is almost flat between 10 m and 60 m, compared to 60 m and onwards.



(a) *Performance evaluated on test data separated consecutively from the data set.*

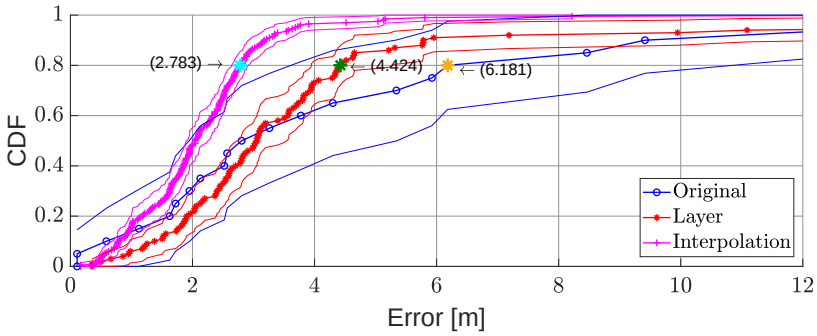


(b) *Performance evaluated on test data separated randomly from the data set.*

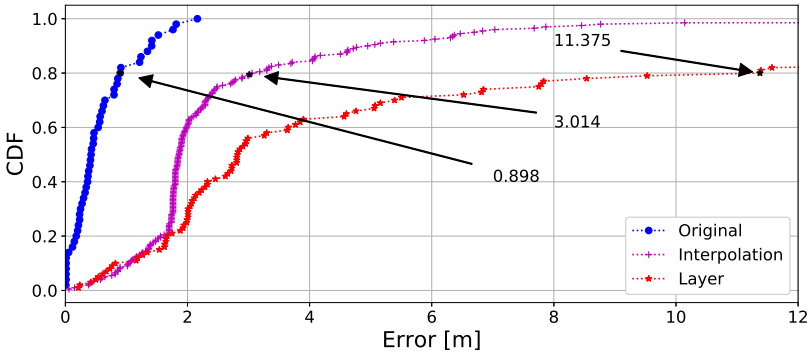
Figure 4.6: *Comparison of the performance for random forests trained and evaluated different data sets. Worth noticing is the low positioning accuracy where the learning and testing set are separated consecutively compared with the learning set and testing set separated randomly. Also notice the good performance for original data set compared with neural networks, Figure 4.2.*

4.3 Positioning in LOS

Positioning in LoS has been done using similar algorithms as in NLoS. For Figure 4.7 the learning and evalation are separated consecutively and in Figure 4.8 randomly.In Figure 4.7a and Figure 4.8a, results for positioning using neural networks in LoS are shown. The neural networks share the same parameters as the ones used for positioning in NLoS. The same parameters as the ones used for positioning in NLoS have been used for both neural networks and random forest. The results of training a random forest on LoS data can be observed in Figure 4.7b and Figure 4.8b.

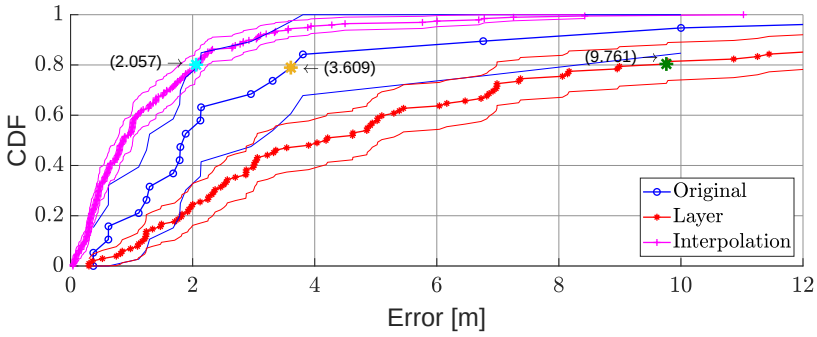


(a) Neural network

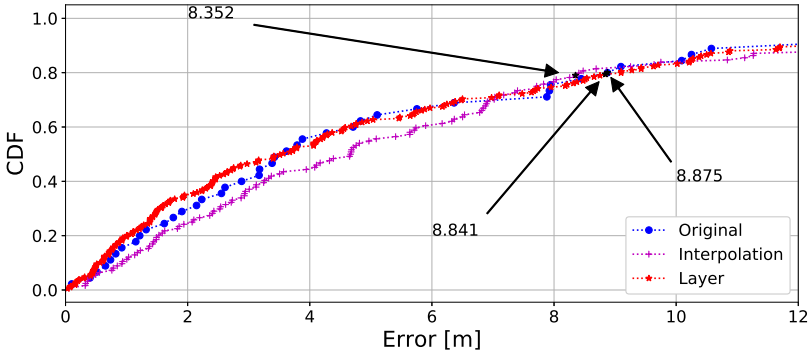


(b) Random forest

Figure 4.7: *Comparison between different positioning algorithm on* LOS *data. The learning and evaluation data are separated consecutively from the data sets in* LOS *conditions. Worth noticing is the better performance for* LOS *conditions compared with* NLOS *conditions.*



(a) Neural network



(b) Random forest

Figure 4.8: *Comparison between different positioning algorithm on* LoS *data The learning data and evaluation data re separated randomly from the data sets in* LoS *conditions. Worth noticing is the better performance for* LoS *conditions compared with* NLoS *conditions.*

By comparing Figure 4.5 and Figure 4.6 with results in Figure **??**, it can be easily observed that the performance for neural networks in LoS are much better there preformance in NLoS, even as good or better than uncertainty for GPS, while for random forest the performances are similar between LoS and NLoS. This might be a consequence of neural networks build models for data, which is easier done in LoS conditions, while random forest classifies by matching data with similar features.

4.4 Detection of NLOS

In Figure D.1, the NP detectors using the difference in BRSRP and DoD are shown. It is marked on the map where there is NLoS. This is done combining where there are buildings and what range the TP has. In Figure 4.9 there is a similar figure with the GLRT using difference in BRSRP and DoD. Both the detection a NP detector and a GLRT gave similar results. The probability of detection P_D and probability of false alarm P_{FA} are calculated. Using signal consisting of difference in BRSRP the $P_D = 88\%$, and using difference in DoD $P_D = 76\%$. For both signals $P_{FA} = 5\%$.

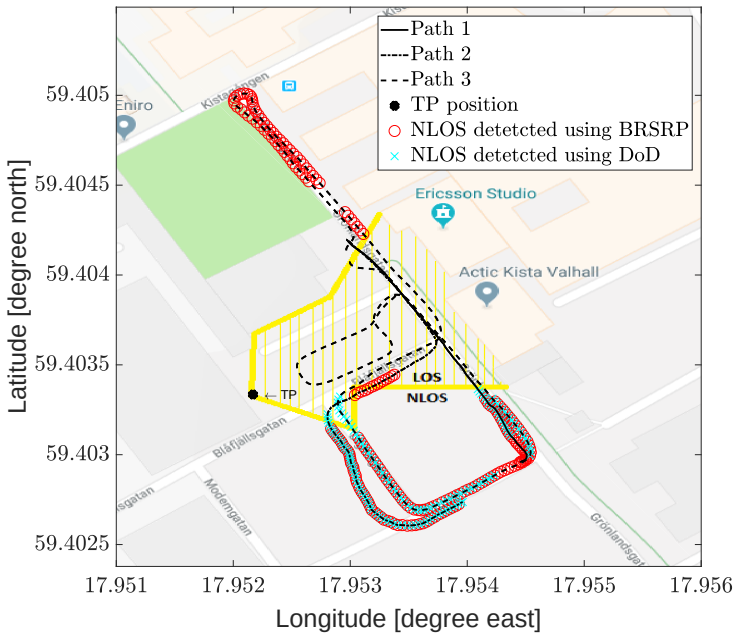


Figure 4.9: *Detection of NLoS using both signals consisting of difference in BRSRP and DoD. The yellow areas are the true classification of LoS conditions for the given TP. This is calculated studying the map with knowledge about the surroundings.*

Worth noticing is that detection using DoD is missing big parts in the end of path 3, in the north west corner. This might be due to that the height of the blocking building is small, so the signal can almost pass over or through the building, which lead to a big part not being detected as NLoS.

Figure 4.10 shows the detector, as applied to path 1. Here both the signals $x_1[n]$ and $x_2[n]$ are shown as well as the NP detector and GLRT for both signals.

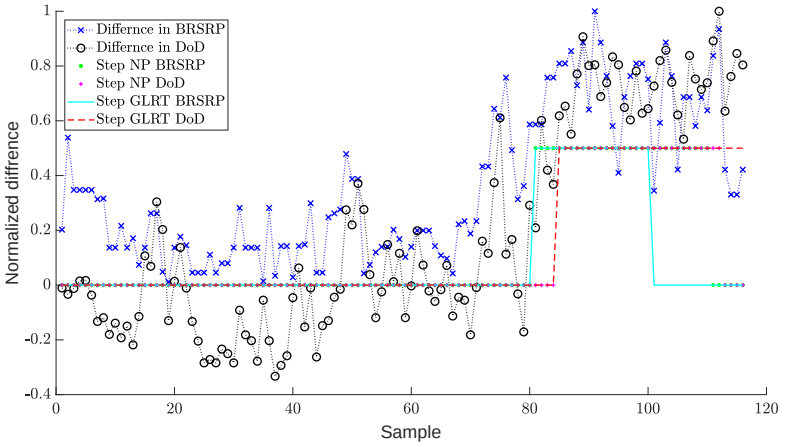


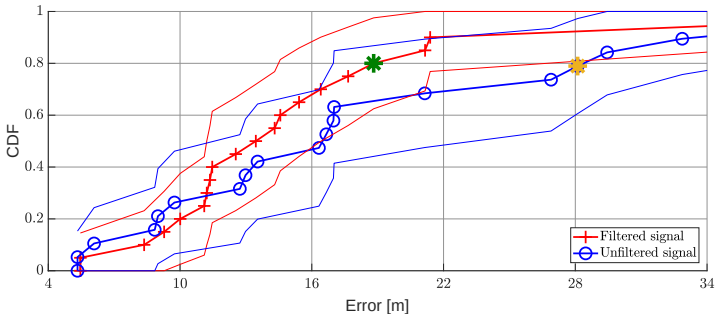
Figure 4.10: *Signals used for detection of NLoS along path 1.*

One can see that two signals look similar and that the NP detector and the GLRT give identical detection results. For more plots corresponding to path 2 and 3 see appendix D.

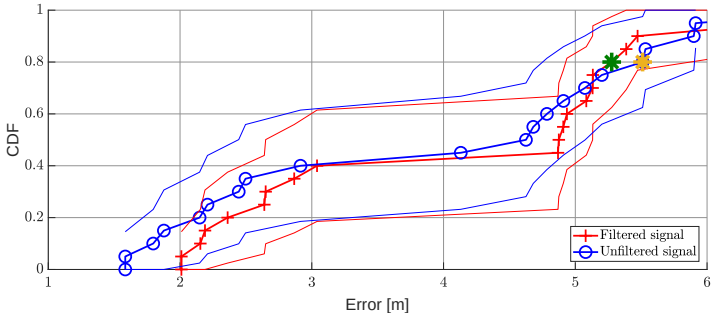
4.5 Kalman Filter

Figure 4.11 shows the result of using a Kalman filter on the output of the of machine learning algorithms. The CDFs of the filtered and unfiltered signals are shown with a 95% confidence interval calculated using *Greenwood's Formula*. The set-up of the filter and the state-space model are described in Section 3.5.

In Figure 4.11a output from the neural networks are filtered with a Kalman filter, and in Figure 4.11b output from the random forest are filtered. The test data set to the machine learning algorithms are consecutively separation of learning and testing set of the original data set. From Figure 4.11 is clear that filtering the output from the neural networks boosts the performance of positioning while for random forest there is no significant improvement. The cause of this can be seen by studying the effect of filtering and the output from the machine learning algorithms seen in Figure 4.12.



(a) *Kalman filter applied to the output from the neural networks.*



(b) *Kalman filter applied to the output from the random forest*

Figure 4.11: *Results of Kalman filtering the output of the machine learning algorithms. It is only the output from machine learning algorithms evaluated on testing set from the original data set that are separated consecutively that are filtered. This due to that for filtering you want to have some consecutive time samples for the filter to track.*

In Figure 4.12 y_{true} is the true distance from the TP, y is the output from the machine learning algorithms, x_f the filtered signal, and x_p the predicted signal from the Kalman filter. It is clear that the estimation of position using random forest is biased, while the estimation in neural networks is not. The state-space model used for the Kalman filter can not take care of the bias, hence no improvement of the accuracy in position.

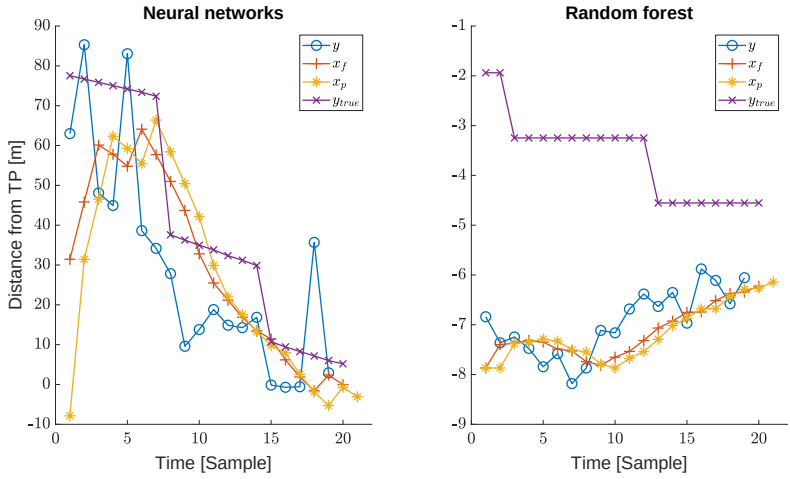


Figure 4.12: *Effect of filtering the machine learning algorithms with a Kalman filter. It i the vertical distance from* TP *to the* UE *on the y-axis. A negative value indicates that the* UE *are south of the* TP *and a positive value that the* UE *are north of the* TP*.*

5

Discussion and Conclusions

This chapter wraps up the thesis, discusses the obtained results in chapter 4, concludes the thesis by looking back on the purpose and problem formulation, and suggests future work.

5.1 Discussion

This section discusses and analysis the results obtained in chapter 4.

5.1.1 Neural networks

From Figure 4.5 one can see that using interpolation of data for learning provides the best results; this with a performance of positioning accuracy below 10 m. This result is for randomly separation of learning and training set (see Figure 4.3). Comparing results obtained with the original data set and the interpolated data set shows that randomly separation of test data set gives better performance compared to evaluating on testing and learning set separated consecutively. With that said, only using a data set with interpolated data have a better performance in positioning accuracy compared with tests evaluated on the original data set. It is also worth mentioning again that the uncertainty of GPS that are in best case around few meters and the grid resolution of the samples is 0.19 m, see Subsection 3.1.1.

Studying the gap in accuracy between evaluation on learning data and testing data in Figure 4.2-4.4 (the difference between the green and solid lines), conclusions regarding the optimal performance and if the neural networks are overfitted or not can be drawn. A small gap indicates that the network is close to its optimal performance and not overfitted. When averaging the position estimation over 100

neural networks the gap between test and learning performance is small. This both when the testing and learning set are separated randomly and consecutively. It gives insights that for randomly separated data, the neural network generalize position estimation.

One more observation is that if the neural networks would have more data to use for training the performance could be improved significantly. This comes from that for evaluation on set of data that contains interpolation of data gives such a good performance for test data separated randomly. The data set with interpolation in combination with random separated replicate a scenario in which a larger data set is available.

5.1.2 Random forest

From Figure 4.6 one can see that the original data gives the best performance using random forest algorithm. The performance evaluated on the original data set is much better for the random forest compared to the neural network. Overall, the random forest gives a much better performance than neural networks. In addition to that training of the random forest is faster than training of the neural network. The random forest takes a couple of hours to train while the neural network takes days.

Some interesting results can be seen in Subsection 4.2.1 and Appendix C, one can see that for all data sets one feature stands out in its importance. Difference in BRSRP between the strongest and some consecutive beam in the antenna are the most inportant feature for all data set except the data set consisting of interpolated data points. One can also see that almost all features are used equally which indicates that all are useful for the machine learning algorithms.

5.1.3 Detection of NLOS

The detection algorithms agree with where NLOS is expected to be, which can be decided by studying the map and environment. This indicates a good choice of signal to use for detection. In Figure 4.10 one can see that the NP detector and the GLRT give identical detection which indicates that the knowledge of the amplitude is redundant.

The probability of detection (P_D) is high while the probability of false alarm (P_{FA}) is low which indicates that the detection algorithms works well. With this said, the P_D and P_{FA} are far from as good as the theoretical values given in Subsection 3.4.2. This indicates that the model is not ideal. It should also be pointed out that the true classification is done very intuitively by studying the map and surroundings and where buildings should block the signal. This leads to the fact that the true classification is very uncertain.

5.1.4 Kalman filter

From Figure 4.11a it is clear that the use of a Kalman filter in combination with a machine learning algorithm can boost the positioning accuracy. Nevertheless, as seen in Figure 4.11b this might not always be the case. In this case it is due to a bias in the output of the random forest which is not accounted for by the state-space model. Increasing the size of the data set would probably solve the the problem with a bias in the estimation. This, since the bias comes from missing data points in the learning set for the machine learning algorithms. When there is no direct match between points in the learning data set and the true target value it might happen that all estimates have bias, this is more likely for a small learning set.

5.2 Conclusions

Going back to the problem formulation in Chapter 1 there are three questions this thesis should have answered. Is it possible to use machine learning algorithms for positioning in urban canyons? Can NLoS and LoS be distinguished? And can filtering improve the performance of the machine learning algorithms?

Results in Chapter 4 shows that machine learning algorithms are great tools for positioning of UEs in a cellular network located in an urban canyon with NLoS conditions. Especially the random forest algorithm got great results despite the limited data set. Since the positioning accuracy of the random forest in urban canyons with NLoS conditions has similar accuracy as the GPS positioning under open sky, which is few meters, some new ground truth measurements might be needed for improving the positioning done by random forest.

Much time have been spent on generation of more data for the machine learning algorithms, some more successful than others. Interpolation using difference in sampling rate of the signal to create new sets of data shows promising results. To confirm that the new set of data replicates the real data, comparison between two data sets of the same size, one with real data and one with a new set of data, have to be made.

One conclusion that can be drawn from generating new sets of data, is that knowledge of the antenna design is useful for positioning with machine learning algorithms. This conclusion is drawn based on how much worse the performance is when considering separation of vertical beam layer in the antenna. It is also possible to say that it is important to separate positioning in LoS and NLoS. This is shown with a better positioning performance for the neural networks in LoS compared with NLoS.

Detection of NLoS works well. Especially using information about BRSRP, both with a NP detector and GLRT. The P_D is almost 90% while the P_{FA} is just a couple of percentages. The drawback of this result is that the true values are

obtained manually, by observing the urban canyon. For development of new NLoS detection algorithms, the outputs of the two studied detectors could be used as the true detection.

To filter the output from the machine learning methods can boost the positioning accuracy. This is true if the state-space model are modelled correctly. Here, for the neural networks the assumption that the system noise was Gaussian distributed with no bias seams reasonable after studying the output, hence the Kalman filter works well. While for the random forest, the bias in the estimation meant that the Kalman filter did not further improve the positioning performance.

5.3 Future Work

Even if the results of this thesis are promising, the data set used for evaluation have been very limited. More data would give the machine learning algorithms a larger learning set which could boost the performance of the positioning methods. A larger data set having more overlapping data points could make it possible to evaluate the machine learning algorithms on consecutively separated testing set that have data points in the learning set with corresponding true positions in the learning set. With longer sequences of estimated positions from the machine learning algorithms the use of Kalman filter will be more relevant since there will be more position to track.

With a larger data set, the problem could even be formulated as a classification problem instead of a regression problem. This would open up for the use of a range of new machine learning algorithms, i.e. k-nearest neighbour. Also trying convolutional neural network (CNN), which has been used to obtain promising positioning accuracy from earlier research, see Section 1.3.

Results in this thesis are achieved with information from only a single TP, but in urban canyons a UE often has connection to multiple TPs. Implementation of positioning methods using multiple TPs is definitely worth further investigation. It would also be interesting to do further investigations on what features are good inputs for the machine learning methods; could the use of more beams be interesting or the change in individual BRSRPs over time? And what makes the last feature in the feature vector have so significant importance in the random forest algorithm (see Subsection 4.2.1 and appendix C).

Developing new detection algorithms to confirm that the LoS and NLoS classifications made in this thesis are of interest. Here, the algorithms used in this thesis could act as the true classification instead of an intuitive one.

Further investigations of combining Kalman filters with machine learning algorithms might be of interest. More time can be spent on tuning the filter and to further study the state-space model.

Appendices

A

Estimation Error

The mean or bias of the estimation error in the x and y directions are numerically calculated for the different estimators as given in (A.1). The estimator $\widehat{\mathbf{pos}} = (\widehat{pos}_x, \widehat{pos}_y)$ is given by (2.2) for the neural network and by definition of random forest in subsection 2.1.2 for random forest.

$$\mathbb{E}(\widehat{\mathbf{pos}} - \mathbf{pos}^{true}) \approx \frac{1}{N}\sum_{i=1}^{N}(\widehat{\mathbf{pos}}_{ij} - \mathbf{pos}_{ij}^{true}) \tag{A.1}$$

Here we have $i = 1 \dots N$ measurements, $j = 1, 2$, denotes the x and y directions and $\mathbb{E}$ denotes the expected value. The results are given in Table A.1, where $\mathbf{b} = (b_x, b_y)$ denotes the estimator error bias. The estimation results are based on a test data set separated with randomly.

Table A.1: *Bias and variance of the estimation error for the different estimators and different learning sets.*

	Neural Network	**Random Forest**
$\mathbf{b}_{\text{Original}}$	$(4.718\ ,\ -2.656)$	$(2.777\ ,\ 0.0174)$
$\mathbf{b}_{\text{Interpolation}}$	$(1.447\ ,\ -0.146)$	$(0.225\ ,\ -0.043)$
$\mathbf{b}_{\text{Layer}}$	$(3.334\ ,\ 1.815)$	$(-0.410\ ,\ -0.193)$

The bias of the estimators in Table A.1 are around zero and small. This lets us conclude that the estimators from the machine learning methods are almost unbiased.

B

Summarized Results

In this chapter the positioning results of the machine learning algorithms are summarised in Table B.1. Worth noticing is that the data sets consisting of interpolated data with randomly separation of learning ad testing set gives best performance in NLOS for both neural network and random forest. Also notice the good performance for random forest on the original data set in NLOS conditions. The performance is measured as at what value the CDF of the position error reach 0.8. The highlighted values are the best performance for different data sets and different ways to separate learning and testing set.

Table B.1: *The performance of positioning using machine learning algorithms.*

		Neural Network		**Random Forest**	
Set of data	**Sampled**	NLOS	LOS	NLOS	LOS
Original	Randomly	22.274 m	3.609 m	11.386 m	8.841 m
Original	Consecutively	26.459 m	6.181 m	**9.525 m**	**0.898 m**
Interpolation	Randomly	**9.242 m**	2.057 m	8.352 m	8.352 m
Interpolation	Consecutively	28.159 m	4.434 m	79.591 m	3.014 m
Layer	Randomly	25.284 m	9.761 m	17.932 m	8.875 m
Layer	Consecutively	39.333 m	2.783 m	47.246 m	11.375 m

C

Feature Importance

In Table C.1 and C.2 the feature importance for the data set containing interpolation between measurements and the data set in which different layers are treated separately are shown. The highlighted features are those with the highest importance.

Table C.1: *The ranking of feature importance generated by the random forest algorithm for data set with interpolation between measurements.*

Best beam nr.	BRSRP	DoD	DDoD	DBRSRP
1	0.0119	**0.1909**	-	-
2	0.0315	0.0070	0.0014	0.0022
3	0.0100	0.0064	0.0016	0.0019
4	0.3698	0.0116	0.0037	0.0946
5	0.0280	0.0036	0.0023	0.0031
6	0.0156	0.0030	0.0028	0.0082
7	0.0331	0.0014	0.0016	0.0042
8	0.0099	0.0009	0.0017	0.0148
9	0.0103	0.0010	0.0010	0.0061
10	0.0318	0.0009	0.0020	0.0135

Table C.2: *The ranking of features importance generated by the random forest algorithm for data set where different layers are treated separately.*

Best beam nr.	BRSRP	DoD	DDoD	DBRSRP
1	0.0147	0.1437	-	-
2	0.0163	0.0703	0.0992	0.0120
3	0.0132	0.0223	0.0090	0.0273
4	0.0125	0.0125	0.0142	0.1286
5	0.0292	0.0113	0.0193	**0.3441**

D

Detection of NLOS

Figure D.1 shows detection of NLOS using a NP detector. The yellow area is marking where the is NLOS conditions.

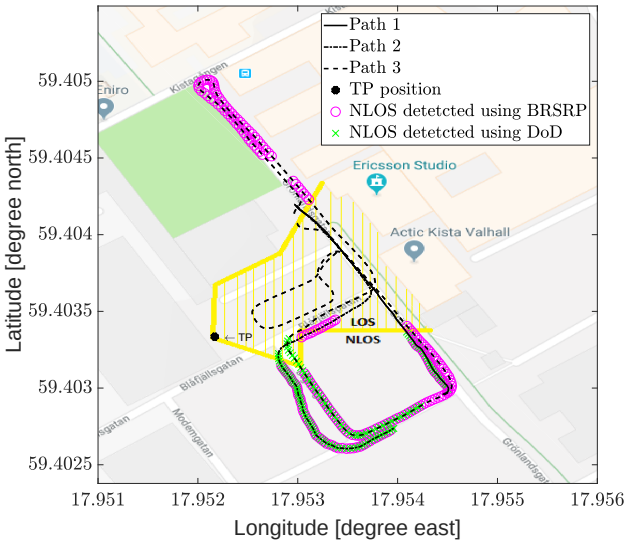
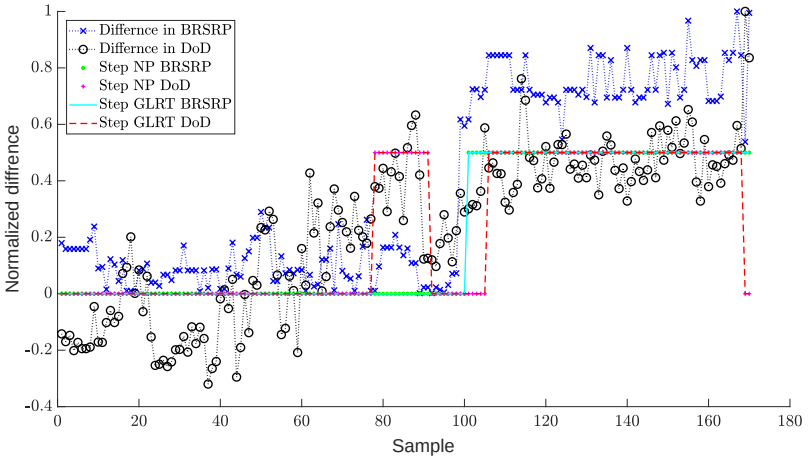
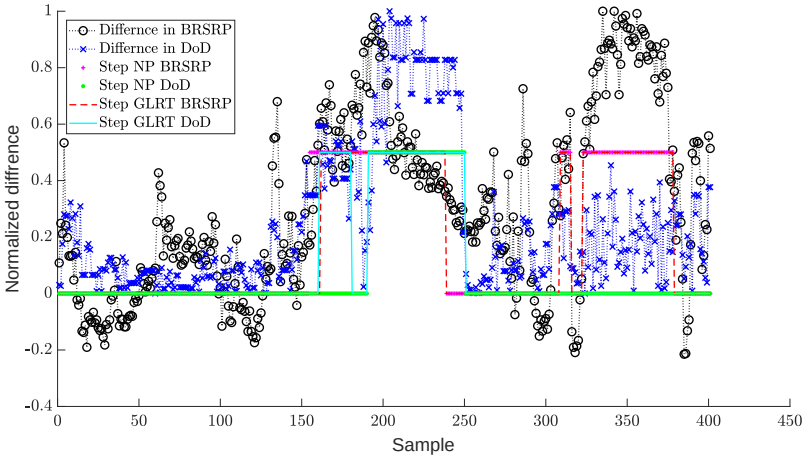


Figure D.1: *Detection of NLOS with an NP detector using both signals consisting of difference in BRSRP and DOD. The yellow areas are the true classification of LOS conditions for the given TP. This is calculated studying the map with knowledge about the surroundings.*

In Figure D.2 the signals used for detection of NLOS on path 2 and 3 are shown. Here both the signals $x_1[n]$ and $x_2[n]$ are shown as well as the NP detector and GLRT for both signals. One can see that two signals look similar and that the NP detector and the GLRT give identical detection results. For more plots corresponding to path 2 and 3 see appendix D. One can also see that classifications using DOD are missing parts between 80 and 100 in path 2 and between 300 and 400 in path 3.



(a) *Signal used for detection of NLOS path 2.*



(b) *Signal used for detection of NLOS path 3.*

Figure D.2: *Signals used for detection of NLOS path 2 and 3.*

Bibliography

[1] T. Roos, P. Myllymaki, and H. Tirri. A statistical modeling approach to location estimation. *IEEE Transactions on Mobile Computing*, 99(1):59–69, August 2002.

[2] X. Zhang, S. M. Razavi, F. Gunnarsson, K. Larsson, J. Manssour, M. Na, C. Choi, and S. Jo. Beam-based vehicular position estimation in 5G radio access. *Special issue in "Localization in Current and Emerging Networks", WCNC*, 2018.

[3] M. Mohannaa, M. Rabehb, E. Zieurb, and S. Hekalab. Optimization of MUSIC algorithm for angle of arrival estimation in wireless communications. *NRIAG Journal of Astronomy and Geophysics*, 2013.

[4] Y. Zhao. *Position Estimation in Uncertain Radio Environments and Trajectory Learning*. Linköping studies in science and technology. Thesis No. 1172, 2017.

[5] J.N. Ash and Lee C. Potter. Sensor network localization via received signal strength measurements with directional antennas. *Proceedings of the 2004 Allerton Conference on Communication, Control, and Computing*, 2004.

[6] J. Vieira, E. Leitinger, M. Sarajlic, X. Li, and F. Tufvesson. Deep Convolutional Neural Networks for Massive MIMO Fingerprint-Based Positioning. *Accepted in the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2017.

[7] J. Borris, P. Hatrack, and N. B. Manclayam. Decision theoretic framework for NLOS identification. *IEEE, Vehicular Technology Conference*, 1998.

[8] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P.-J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on Signal Processing*, 50(2):425 – 437, February 2002.

[9] F. Gunnarsson, F. Lindsten, and N. Carlsson. Particle filtering for network-based positioning terrestrial radio networks. *ET Conference on Data Fusion & Target Tracking*, 2014.

[10] F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22, July 2005.

[11] W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 52(1-2):99–115, 1990.

[12] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[13] D. J. Livingstone. *Artificial Neural Networks Method and Applications*. Hummana Press a part of Springer Science + Business Media LCC, 2008.

[14] S. M. Kay. *Fundamentals of Statistical Signal Processing Volume 1: Estimation Theory*. Prentice Hall PTR, 1993.

[15] L. Breiman. Bagging predictors. *Springer Link Machine Learning*, 24:123–140, 1996.

[16] A. Liaw and M. Wiener. Classification and regression by random forest. *R News*, 2002.

[17] L. Breiman. Random forests. *Springer Link Machine Learning*, 45:5–32, October 2001.

[18] S. M. Kay. *Fundamentals of Statistical Signal Processing Volume II: Detection Theory*. Prentice Hall PTR, 1998.

[19] F. Gustafsson, L. Ljung, and M. Millnert. *Signal Processing*. Studentlitteratur, 2011.

[20] GPS Accuracy. Technical report, National Coordination Office for Space-Based Positioning, Navigation, and Timing, 2017. Information collected 5 April 2018.

[21] 3GPP Ts 38.215 v15.1.0 technical specification group radio access network NR Physical layer measurements (Release 15). 2018-03.

[22] 3GPP TR 37.857 v13.1.0 study on indoor positioning enhancements for UTRA and LTE (release 13).

[23] A.W. van der Vaart. *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridege University Press, 1998.

[24] S. B. Cho and J.H. Kim. Combining multiple neural networks by fuzzy integral for robust classification. *IEEE Transactions on Systems, Man, and Cybernetics*, 1995.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research 15*, 2014.

[26] Z. Zhou, J. Wu, and W. Tang. Ensembling neural networks: Many could be better than all. *Elsevier Artificial Intelligence*, 2002.

[27] K. Kuźniar and M. Zając. Some methods of pre-processing input data for neural networks. *Computer Assisted Methods in Engineering and Science*, 22:14–1511, 2015.

[28] T. M. Oshiro, P. S. Perez, and J. A. Baranauskas. How many trees in a random forest? *Machine Learning and Data Mining in Pattern Recognition 8th International Conference, MLDM 2012 Springer*, pages 154–168, 2012.

[29] F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.