Czech Technical University in Prague
Faculty of Electrical Engineering

BACHELOR THESIS

Libor Urbaník

Methods for People Localization Based on Radio Signals

Department of Measurement
Supervisor: **Ing. Libor Přeučil, CSc.**

Prague, 2014

CZECH TECHNICAL UNIVERSITY IN PRAGUE
Faculty of Electrical Engineering
Department of Measurement
Technická 2, 166 27 Prague 6, Czech Republic

Academic year **2013/2014**

BACHELOR PROJECT ASSIGNMENT

Student: **Libor Urbaník**

Study programme: **Cybernetics and Robotics**
Branch: **Sensors and Instrumentation**

Title of Bachelor Project: **Methods for People Localization Based on Radio Signals**

Guidelines:

1. Study the problem of object/people localization based on radio signals. Assume moving objects/people equipped with radio tags, that communicate with a base station. The locations of the base stations are known.
2. Implement selected methods for patient localization based on quality of radio signals (e.g. K-means, Nearest Neighbor, etc.).
3. Extend the previous methods to consider dynamics of the objects/people (e.g. using object tracking and a known map of the environment).
4. Create datasets with HW provided by the supervisor.
5. Experimentally verify the implemented methods, compare precision of the localization with and without consideration of dynamics of the objects.

Bibliography/Sources:

[1] Thrun, S.,Burgard, W., Fox, D.: Probabilistic robotics, MIT Press, 2005
[2] Kelly, A.: Mobile robotics, mathematics, models and methods, Cambridge university press, 2013
[3] Kuai, X., Yang, K., Siyao, F., Zhen, R.: Simultaneous localization and mapping (SLAM) for indoor autonomous mobile robot navigation in wireless sensor networks, In proceedings of International Conference on Networking, Sensing and Control, 2010

Bachelor Project Supervisor: Ing. Libor Přeučil, CSc. (K 13133)

Valid until: the end of the summer semester of academic year 2014/2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
Fakulta elektrotechnická
L.S.
-4-

Prof. Ing. Vladimír Haasz, CSc.
Head of Department

Prof. Ing. Pavel Ripka, CSc.
Dean

Prague, February 18, 2014

Annotation

In hospitals it is very important, that if a patient is in a need, a medical personnel can reach him in the shortest possible time. This time can be further decreased by sending a closest doctor/patient to help the patient. Therefore, a system for personnel and patients localization in hospitals could increase the quality of the provided health care. For the purposes of localization, a network of receivers was installed in a hospital. This thesis aims to investigate, how to use the prepared system for people localisation inside of the hospital. Firstly, measurements were performed to determine the parameters of this system. Afterwards several methods of dataset creation were tested and compared by their suitability for purposes of a dynamic object localization in a known environment. Further, two classification methods—k-Nearest neighbour and C4.5—were implemented for the task of personnel localization. These methods were compared on a static and dynamic localization task. We verified, that it is possible to determine a person's position using mentioned classification methods.

Anotace

V nemocnicích je velmi důležité, aby zdravotní personál byl schopný se dostat k pacientovi v nouzi v co nejkratším čase. Tento čas může být zkrácen, když se na místo pošle nejbližší doktor/pacient aby mu pomohl. Z tohoto důvodu by mohl systém pro lokalizaci pacientů a personálu zvýšit kvalitu poskytované zdravotní péče v nemocnicích. Pro účely lokalizace byla v nemocnici nainstalována síť přijímačů. Cílem této práce je zjistit, jak využít připravený systém pro účely lokalizace lidí v nemocnici. Nejdříve byla provedena měření k určení parametrů tohoto systému. Následně bylo vyzkoušeno několik metod vytváření datasetů, které byly porovnány pro zjištění použitelnosti na problém dynamické lokalizace ve známém prostředí. Dále byly implementovány dvě klasifikační metody pro úlohu lokalizace osob: k-nejbližších sousedů a algoritmus C4.5. Tyto metody byly porovnány na úlohách statické a dynamické lokalizace. Ověřili jsme, že je možné určit pozici osob použitím zmíněných klasifikačních metod.

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostaně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ....
Podpis autora práce

Acknowledgements

I would like to thank my supervisor Ing. Libor Přeučil, CSc. and Ing. Vojtěch Vonásek for a great help and many useful advises during the work on this thesis.

Moreover, I would like to thank my family for their never-ending support throughout my life.

Contents

Chapter 1

Introduction

In hospitals it is very important, that if a patient is in a need, a medical personnel can reach him in the shortest possible time. This time can be further decreased by sending a closest doctor/patient to help the patient. Therefore, a system for personnel and patients localization in hospitals could increase the quality of the provided health care.

In the last few decades, many radio signal-based systems were used for the task of outdoor localization. For the indoor environment, however, these systems are unusable due to the attenuation of radio signals in walls [1].

The goal of this thesis is to investigate how to localize patients and employees in hospitals using radio signals. In this hospital, unknown network of receivers (Fig. 1.1) is installed. The purpose of these receivers is to collect information transmitted by the transmitters (Fig. 1.2) nearby. The transmitters are carried by patients and the hospital personnel. Since the properties of the used system are unknown, we will have to design and perform experiments in the university to determine capabilities of this system. Thereafter, we will choose several methods for this task and compare them on an experimental receiver network, composed of three and six receivers.

Figure 1.1: Receiver

Figure 1.2: Transmitter

Chapter 2

Problem analysis

Objects to be localized are matchbox-sized transmitters carried by the hospital personnel. They're using surface mount module from the JN5148-001-Myy family, based on wireless microcontroller from Jennic. The module is targeted at JenNet and ZigBee PRO networking applications [2].

Each transmitter is broadcasting it's own identification code and a button state indicator. This information—when captured by a receiver—is stored together with a signal quality value, computed by the link quality indicator(LQI). According to the IEEE 802.15.4 standard [3] (section E.2.3), "The LQI measures the received energy level and/or SNR for each received packet". The algorithm is not described in detail. However, in the IEEE 802.15.4 standard [3] (section 6.7.8) are specified properties of computed signal quality values:

> The minimum and maximum LQI values (0x00 and 0xff) should be associated with the lowest and highest quality IEEE 802.15.4 signals detectable by the receiver, and LQ values in between should be uniformly distributed between these two limits. At least eight unique values of LQ shall be used.

The fact that the chip provides signal quality values brings us to idea of people localization using these devices. In this thesis, we will investigate whether the objects (patients) can be localized using the measured signal quality values. However, due to lack of information provided by the manufacturer, we don't aim to estimate exact position of people. Rather, we will try to determine in which room/area a person is located. In the example 2.1 below, a few lines of the acquired data are shown. Each line consists of eight parts. The known parts are: the first, which contains a time of receipt in seconds; the fourth, which is an identification number of the receiver;

the fifth containing an identification number of the transmitter; the seventh representing measured signal quality in a hexadecimal format; and the last number, which denotes, whether the button was pressed (1) or not (0). In the experiments later on, the measured signal qualities will be converted from hexadecimal to decimal format, prior to depicting in a figure.

```
1218184546.172808  D  LQ  00AA  004E  18CF  5D  0
1218184546.212738  D  LQ  00A4  004E  18C3  5A  1
1218184546.228053  D  LQ  009C  004E  18CE  30  1       (2.1)
1218184546.248856  D  LQ  00A4  004A  BB4F  4F  0
1218184546.264449  D  LQ  00AA  004A  D327  57  0
```

Since there is no detailed description of the used system, we firstly had to analyse its properties. Especially we tried to investigate, how the directional orientation of the transmitter influences the measured signal quality and how the measured signal quality depends on the distance between the transmitter and the receiver.

2.1 Transmitter

For the purposes of people localization it is desired, that the measured signal quality is independent on the directional orientation of the transmitter. This property is important, as the transmitters are carried by people and therefore it's not possible to ensure a same orientation towards the antennas at all time. The radiation patterns shown in Fig. 2.1 were measured in three positions of the transmitter, one meter far from the receiver. In each position, the transmitter was rotated around its centre by 45 degrees and the signal quality was measured for two minutes after each rotation[1]. The measured values were averaged and depicted in a polar plot. The measured plots are roughly symmetrical, however, the values in the second position are slightly lower. The reason is unclear, but it could be caused by the greater contact surface of the transmitter with the plastic pad.

2.2 Receiver

The purpose of the receivers is to measure signal quality of the data transmitted by transmitters. Our goal is to determine, if the signal quality values are independent on the position of receiver antennas, which is an important

[1] Approximately 120 values for each point.

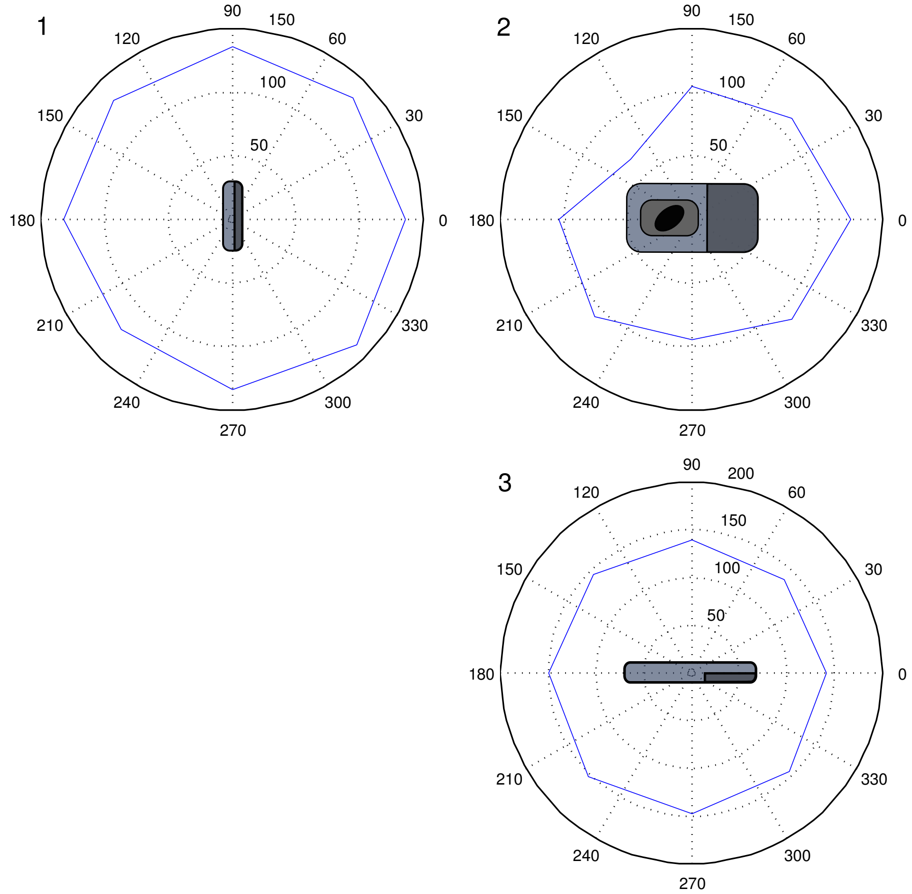

Figure 2.1: Radiation patterns of the transmitter

property, because it's not possible to maintain the same position of these antennas all the time.

One of the tests performed was aimed to determine if the position of receiver antennas have an influence on the measured signal quality. Two transmitters were placed in a 0.2 and 4 meter distance from the receiver and the signal quality was measured for approximately one minute. In Fig. 2.2, the measurements for two positions of the antennas are shown. In the first measurement, the antennas were pointing towards transmitters and in the second, upwards. The position of antennas affected measured values; the signal quality from the closer transmitter was higher, when the antennas were pointed at it, but the signal quality of the further transmitter was lower. The difference in the measured values weren't significant, which is convenient for our task.

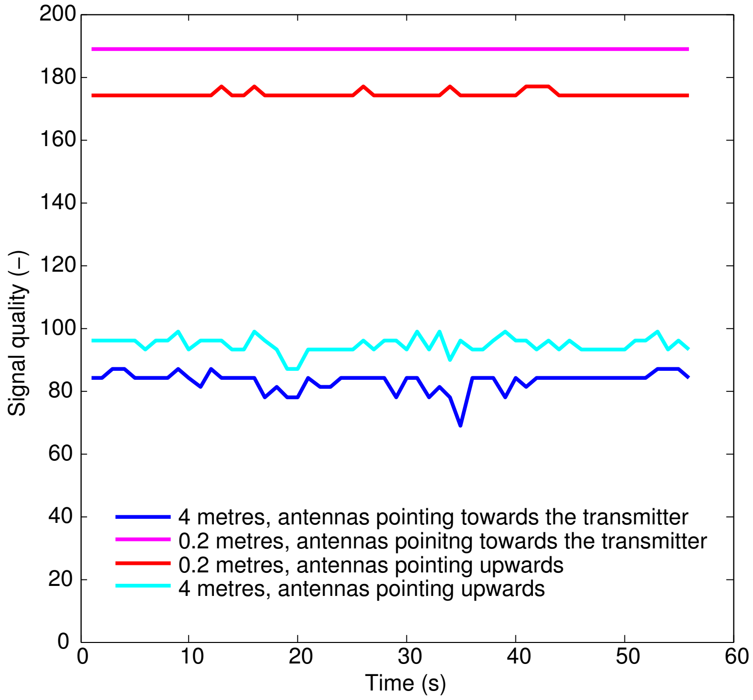

Figure 2.2: Influence of the receiver antenna position on the signal quality

2.3 Single-receiver tests

In the next test, the signal quality was measured for five different distances between the transmitter and the receiver in a corridor(Fig. 2.3). Measurement in each position lasted two minutes.

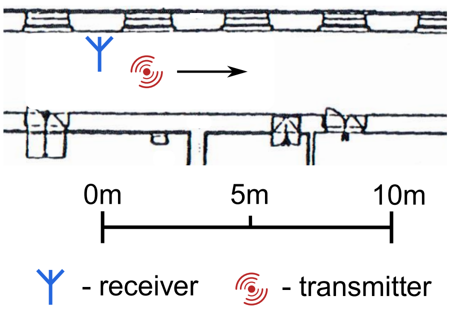

Figure 2.3: Setting for the experiment

The influence of the distance between the receiver and the transmitter on the signal quality is shown in Fig. 2.4. The signal quality was in most cases decreasing with the increasing distance. This measurement indicates, that the signal quality can be used to localize people.

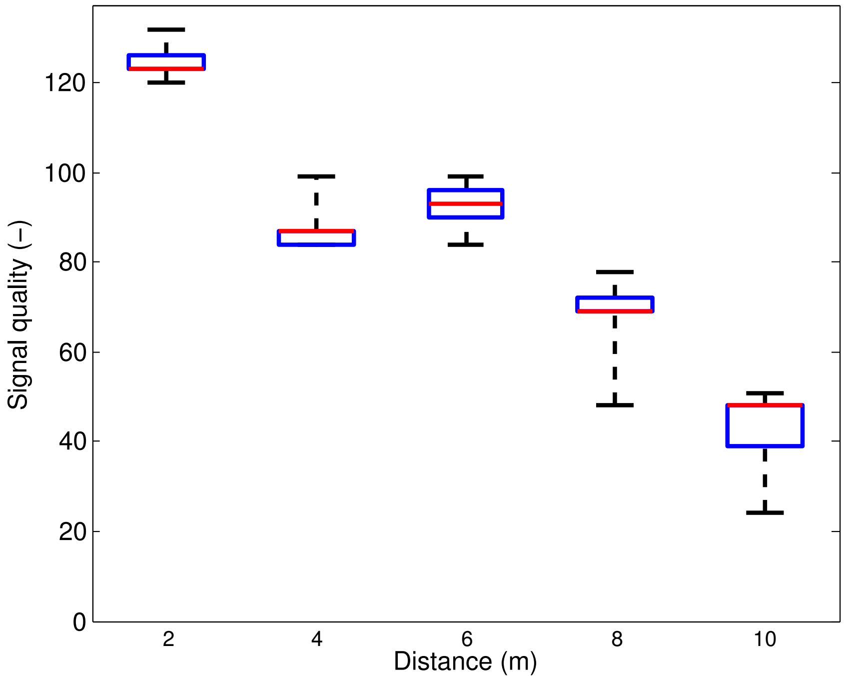

Figure 2.4: Influence of the distance between the receiver and the transmitter on the signal quality

2.4 Multiple-receiver tests

The following experiment has taken place in a corridor with the transmitter placed in the middle and the receivers placed as shown in Fig. 2.5. Firstly, the corridor was empty and the signal quality was measured for one minute. Secondly, there were people walking in the corridor, to better emulate real conditions in hospitals.

Fig. 2.6 shows, that human activity in the way of the signal from the transmitter to the receiver increases standard deviation of the signal quality. Furthermore, it may increase the mean value.

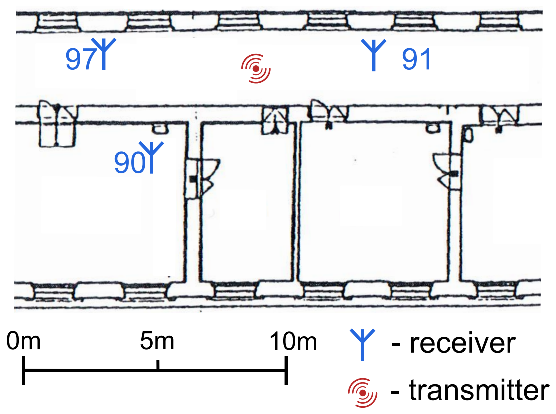

Figure 2.5: Setting for the experiment

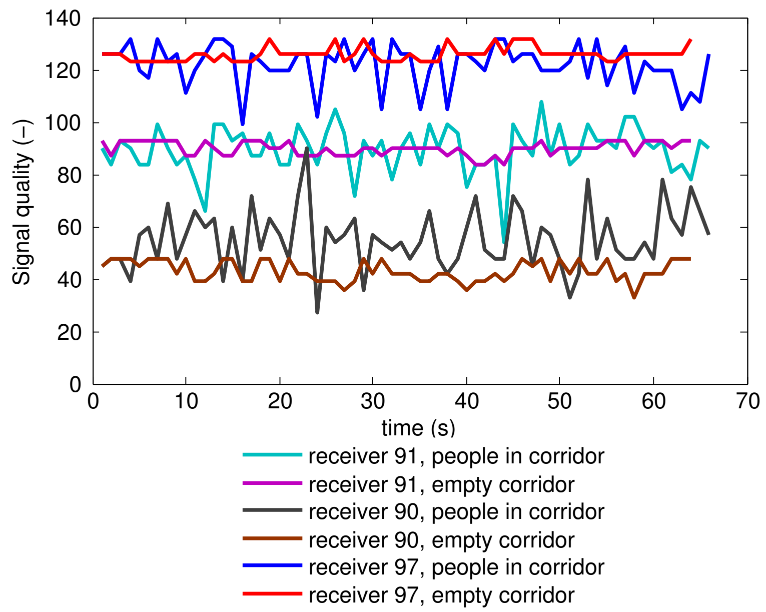

Figure 2.6: Influence of the person walking around transmitter on the signal quality

2.5 Conclusion

Previous measurements show, that neither orientation of the transmitter, nor the position of the receiver antennas have significant influence on the measured signal quality. This could make the dependency of the signal quality on the distance between the transmitter and the receiver useful for a localization of hospital personnel.

Chapter 3

Classification methods

The purpose of the used system is to determine a room or set of rooms, in which a person is located, based on the collected data. This means to assign a class $c \in C$ to a data vector $\mathbf{Q} \in \mathbb{R}^n$, where C is a set of all rooms and $\mathbf{Q}$ contains measured signal qualities from n receivers. The problem of assigning a class c to a data vector $\mathbf{Q}$ is a classification problem, which has been studied for many years in artificial intelligence and machine learning [4].

There are many classification methods. For this task, we selected the k-nearest neighbours algorithm and C4.5 algorithm.

3.1 k-Nearest neighbours

First of the proposed methods is the k-nearest neighbours algorithm(kNN). In this case, the class c of the input data vector $\mathbf{Q}$ is determined by the class of the majority of k-nearest neighbours in the provided training set. To find these nearest neighbours, kNN uses a chosen distance metric. The metric used in our case is Euclidean distance. Fig. 3.1 shows an example of kNN classification in a two-dimensional space for $k = 3$. The k is usually an odd number, so the neighbours from one class always outnumber the others.

3.2 C4.5

C4.5 is an algorithm for generating a decision tree developed by John Ross Quinlan. It is an extension of his earlier algorithm ID3 [5].

Starting with the training data at the root node, C4.5 calculates entropy for each unused attribute[1] and splits the set by the one with the lowest en-

[1] Attributes will be described in Chapter 4

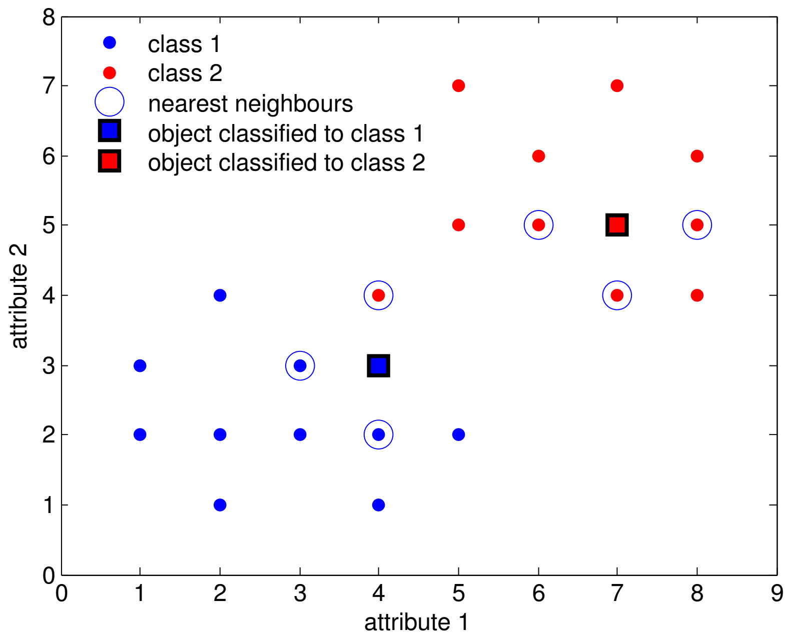

Figure 3.1: Example of a kNN classification for k=3

tropy. A new decision tree node containing the chosen attribute is made and the splitting continues recursively with the created subsets. The recursion stops in one of these cases:

- Each element from the subset belongs to the same class: The terminal node contains a name of the class.
- The subset is empty: The terminal node contains a name of the most frequent class from the parent subset.

Chapter 4

Software used for classification

Experiments were performed in Weka [6], which is an open source software written in Java used for machine learning and data mining tasks[1]. It can be controlled via graphic user interface or command line. Also, it is possible to call its algorithms directly from a Java code.

4.1 Dataset

To employ algorithms implemented in Weka for our task, a proper dataset has to be created. The dataset contains all the information about the input data. In Weka, the whole dataset is implemented as weka.core.Instances class and the individual elements as weka.core.Instance class objects. Each instance consists of predefined types of attributes. There are five types of attributes used in Weka:

- Numeric - a real or integer number
- Integer - treated as numeric
- Real - treated as numeric
- Nominal - one value from a predefined list
- String - containing text values
- Date - date in a specified format
- Relational - for multi-instance data

Attributes are described thoroughly in a Weka manual included in the distribution.

[1] Available at http://www.cs.waikato.ac.nz/ml/weka/index.html (5/2014)

4.2 ARFF file

A dataset can be saved in an ARFF file, which is an external representation of a weka.core.Instances class. The ARFF file consists of two sections. The first section is a header, containing a name of the dataset and a definition of the attributes. The second section, starting with line @data, contains actual data.

In the example 4.1 is a part of an ARFF file, created from data acquired in the experiment shown in Fig. 5.1. The first line of the header section contains @*relation relation_name*, where *relation_name* is a name of the relation. On the following lines, used attributes are defined; the order of these attributes specifies, in which columns of the data section are placed. In this work, we use two types of attributes; numeric and nominal. Numeric attributes are for values measured by receivers. Nominal specify a room c from a set of all rooms C, in which the measurement was taken. The attributes are in a form @*attribute attribute_name value_type*. The *value_type* for numeric attributes is *numeric*, *real* or *integer*. For a nominal attributes, the *value_type* contains a list of possible values enclosed in braces. Weka considers the last attribute as a class attribute, if not specified otherwise.

Using the structure with n attributes described above we achieve, that each line of the data section (each instance of the dataset) contains a data vector $\mathbf{Q} = [attribute_1, \ldots, attribute_{n-1}]$ and a class c of this vector ($attribute_n$).

```
@relation localization

@attribute quality91 numeric
@attribute quality90 numeric
@attribute quality97 numeric
@attribute position {room1, room2, room3}

@data
130 85 37 room1
127 71 41 room1
91 125 83 room2
48 73 119 room3
24 80 138 room3
```

(4.1)

Chapter 5

Experiments

The purpose of experiments in this chapter is to find possibilities of personnel localization using classification algorithms implemented in Weka. Firstly, we aimed to examine, if it's possible to determine a person's position in an office environment, using chosen algorithms J48[1] and kNN. In these tests, three receivers were used. Secondly, we tried to apply our results on a more complex scenario with six receivers.

As shown in the Fig. 2.6, the measured signal quality is fluctuating. We supposed, that suppressing the influence of ambient effects could decrease a number of instances, that our classifiers classified incorrectly. Therefore, a performed tests included classifiers trained and tested either on raw values and processed values. For the processing, functions computing average and median values from measured signal qualities were chosen.

To compare our approaches, we used a percentage of correctly classified instances, i.e. a percentage of data vectors $\mathbf{Q}$, that had been correctly classified. We call it a classifier success rate.

5.1 Dataset preparation

Three receivers were placed in adjacent rooms as shown in Fig. 5.1. A person carrying two transmitters, $\mathrm{T_A}$ in a hand and $\mathrm{T_B}$ in his pocket, remained in each marked area for two minutes. The transmitter $\mathrm{T_A}$ was held in various heights and near walls to get diverse values. Afterwards, these values were used to create a dataset for training classifiers and the values acquired from the transmitter $\mathrm{T_B}$ were used to create a dataset for evaluating these classifiers. Fig. 5.2 shows the difference between measured values from the two transmitters in a position 1 of the experiment from Fig. 5.1. As you can see,

[1]J48 is an implementation of C4.5 algorithm in Weka.

a placement of the transmitter influenced a distribution of the measured signal qualities. However, the median values of these qualities haven't changed as much. Results of this measurement are similar to the findings shown in Fig. 2.6, from the experiment in Section 2.4.

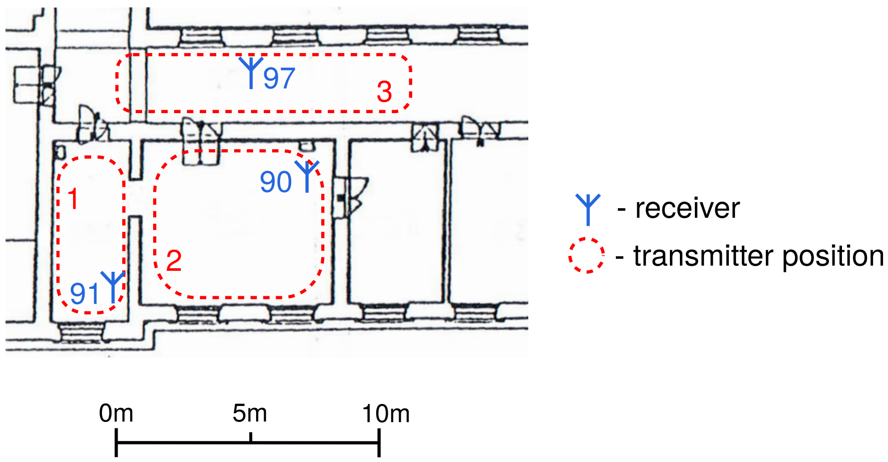

Figure 5.1: Setting for the experiment with three receivers

The data measured by each receiver is a set of signal quality values R^T_{ni}, where n is a number of receiver, i is an index of the measured value and $T \in \{\mathrm{A}, \mathrm{B}\}$ stands for the used transmitter.

We made a comparison of classifiers trained on different datasets to determine, which approach brings higher success rates at determining a person's position. These datasets D_m differed in a method, by which their data vectors were created, where the m denotes a number of processed values. For the purposes of classification, it is needed to create a data vector $\mathbf{Q}$ in a suitable way. In this chapter, we will investigate various approaches to the creation of $\mathbf{Q}$.

5.1.1 Unprocessed values ($\mathbf{D}_{\mathbf{upd}}$)

This is the simplest case, where the i-th data vector $\mathbf{Q}_i \in \mathrm{D}_{\mathrm{upd}}$ contains unprocessed R^T_{ni} values, i.e. $\mathbf{Q}_i = \{q_{ni}\}$, where $q_{ni} = R^T_{ni}$.

5.1.2 Average from x values ($\overline{\mathbf{D}}_x$)

In this case, the j-th data vector $\mathbf{Q}_j \in \overline{\mathrm{D}}_x$ contains averaged R^T_{ni} values, i.e. $\mathbf{Q}_j = \{\overline{q}_{nj}\}$, where

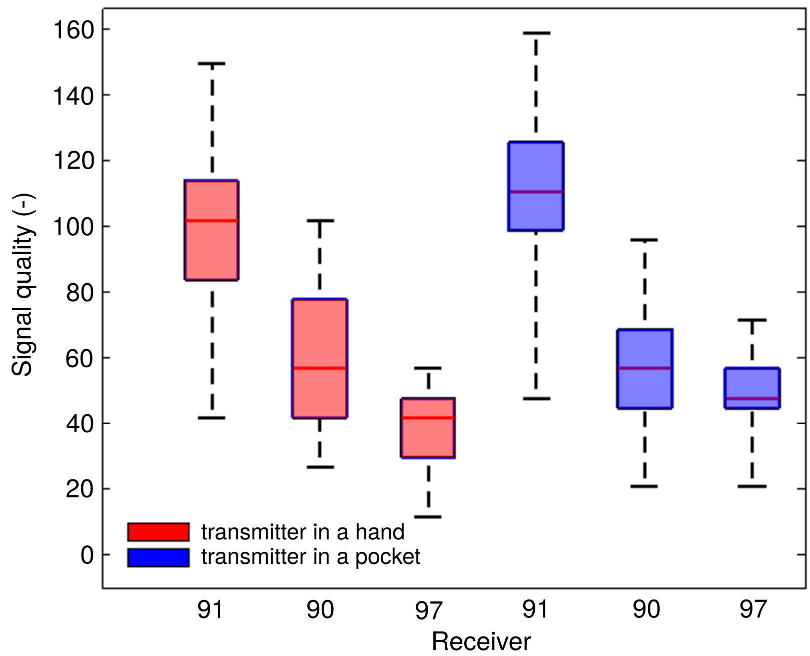

Figure 5.2: Difference between values from two transmitters

$$\overline{q}_{nj} = \frac{\sum_{i=k}^{jx} R_{ni}^{T}}{x}, \text{ where } k = (j-1) \times x + 1. \tag{5.1}$$

In Fig. 5.3, method of selecting $x = 3$ values to process from a measured data is shown.

5.1.3 Average from x values with standard deviations ($\overline{\mathbf{D}}_x^{\prime}$)

In this case, the j-th data vector $\mathbf{Q}_j \in \overline{\mathrm{D}}_x^{\prime}$ contains averaged R_{ni}^{T} values with standard deviations, i.e. $\mathbf{Q}_j = \{\overline{q}_{nj}, \sigma_{nj}\}$, where $\overline{q}_{nj}$ is an average value calculated from equation 5.1 and the deviation σ_{nj} is calculated as

$$\sigma_{nj} = \sqrt{\frac{1}{x} \sum_{i=k}^{jx} (R_{ni}^{T} - \overline{q}_{nj})}, \text{ where } k = (j-1) \times x + 1. \tag{5.2}$$

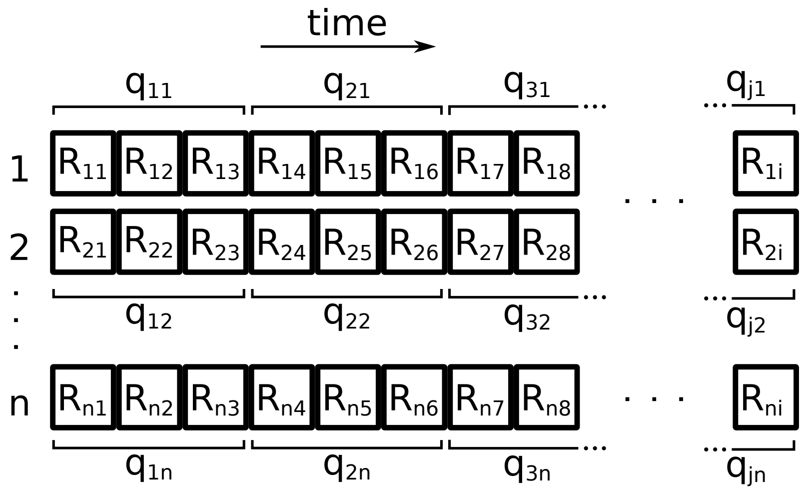

Figure 5.3: Selecting values for creating a vector Q_j

5.1.4 Median from x values ($\tilde{\mathbf{D}}_x$)

This method is the same as the method described in Subsection 5.1.2, but instead of average values, the j-th data vector $\mathbf{Q}_j \in \tilde{\mathrm{D}}_x$ contains median values, i.e. $\mathbf{Q}_j = \{\tilde{q}_{nj}\}$, where $\tilde{q}_{nj}$ is a median calculated from the last x measurements.

5.1.5 Median from x values with standard deviations ($\tilde{\mathbf{D}}_x^{'}$)

This method is the same as the method described in Subsection 5.1.3, but instead of average values, the j-th data vector $\mathbf{Q}_j \in \tilde{\mathrm{D}}_x^{'}$ contains median values, i.e. $\mathbf{Q}_j = \{\tilde{q}_{nj}, \sigma_{nj}\}$.

5.1.6 Comparison

In the following tables, we compared J48 and kNN classifiers. These classifiers were trained on various training datasets *tn*. For evaluation, test datasets *tt* were used. Data vectors in these datasets were created by the methods specified above.

Firstly, we tried to determine, if averaging the values in train or test datasets influences the classification success rates. Data vectors were created from unprocessed values and average values from 5, 9 and 19 measurements. In the following tables, each column is represented by a number of measured signal qualities *tn*, that were averaged to create a training dataset. *tt* specifies a number of averaged values used to create a dataset for evaluating classifiers.

tt \ tn	D_{upd}	$\overline{D}_5$	$\overline{D}_9$	$\overline{D}_{19}$
D_{upd}	85.37%	70.33%	85.37%	88.62%
$\overline{D}_5$	97.92%	81.25%	91.67%	97.92%
$\overline{D}_9$	100.00%	88.89%	92.59%	100.00%
$\overline{D}_{19}$	100.00%	89.23%	100.00%	100.00%

Table 5.1: Success rates of J48 classifier trained and tested on averaged values

tt \ tn	D_{upd}	$\overline{D}_5$	$\overline{D}_9$	$\overline{D}_{19}$
D_{upd}	93.09%	92.28%	91.87%	92.68%
$\overline{D}_5$	100.00%	100.00%	100.00%	100.00%
$\overline{D}_9$	100.00%	100.00%	100.00%	100.00%
$\overline{D}_{19}$	100.00%	100.00%	100.00%	100.00%

Table 5.2: Success rates of kNN classifier trained and tested on averaged values

As shown in Table 5.1 and Table 5.2, both classifiers were able to determine a person's position with 100% accuracy, if values were averaged. In this case, kNN classifier performed better than J48, if the values were averaged from a smaller number of measurements.

Thereafter, we added standard deviations of the averaged values to the datasets used in previous experiment. Table 5.3 shows, that success rates of used classifiers were worsened. However, 100% success rates were achieved.

	J48			kNN		
tt \ tn	$\overline{D}'_5$	$\overline{D}'_9$	$\overline{D}'_{19}$	$\overline{D}'_5$	$\overline{D}'_9$	$\overline{D}'_{19}$
$\overline{D}'_5$	84.75%	94.92%	79.66%	98.31%	93.22%	86.44%
$\overline{D}'_9$	88.89%	92.59%	81.48%	100.00%	96.30%	88.88%
$\overline{D}'_{19}$	89.23%	100.00%	92.30%	100.00%	100.00%	92.31%

Table 5.3: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

tn \ tt	J48 $\tilde{D}_5$	J48 $\tilde{D}_9$	J48 $\tilde{D}_{19}$	kNN $\tilde{D}_5$	kNN $\tilde{D}_9$	kNN $\tilde{D}_{19}$
$\tilde{D}_5$	89.23%	81.54%	86.15%	100.00%	100.00%	100.00%
$\tilde{D}_9$	94.44%	86.11%	86.11%	100.00%	100.00%	100.00%
$\tilde{D}_{19}$	94.44%	94.44%	88.89%	100.00%	100.00%	100.00%

Table 5.4: Success rates of J48 and kNN classifiers trained and tested on median values

tn \ tt	J48 $\tilde{D}'_5$	J48 $\tilde{D}'_9$	J48 $\tilde{D}'_{19}$	kNN $\tilde{D}'_5$	kNN $\tilde{D}'_9$	kNN $\tilde{D}'_{19}$
$\tilde{D}'_5$	93.85%	81.54%	66.15%	100.00%	98.46%	93.85%
$\tilde{D}'_9$	97.22%	86.11%	77.78%	100.00%	100.00%	86.11%
$\tilde{D}'_{19}$	100.00%	94.44%	83.33%	100.00%	100.00%	100.00%

Table 5.5: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

In Table 5.4 and Table 5.5 are shown classification success rates, when median values were used instead of averaged values.

When we used datasets with median values instead of averaged values, the kNN classifier in most cases classified with 100% accuracy. J48 also managed to do this, however, not as often as kNN.

5.1.7 Conclusion

From the performed experiments we can conclude, that both J48 and kNN classifiers were in most cases achieving higher success rates, if the training dataset was created from unprocessed values or from values processed from a lower number of measurements. In this task it is important, that a classifier reliably determine a position from a lower number of measurements. For this reason, the kNN classifier trained on averaged values, median values or median values with standard deviations was more suitable for this task.

The method of creating datasets from the experiments above has a few disadvantages. The first is, that each receiver measures signal quality values roughly once in a second (Fig. 5.4). To create a data vector **Q**, which elements are made from n averaged signal quality values, we have to measure n seconds. This means, that we can determine a person's position only once in a n seconds. The second drawback is, that receivers do not always measure signal quality values at a same rate, as we assumed. If a certain receiver hasn't measured any value in a two seconds, we supposed, that a signal quality at this receiver is minimal. This could increase classification errors. This is why we designed another method of creating datasets.

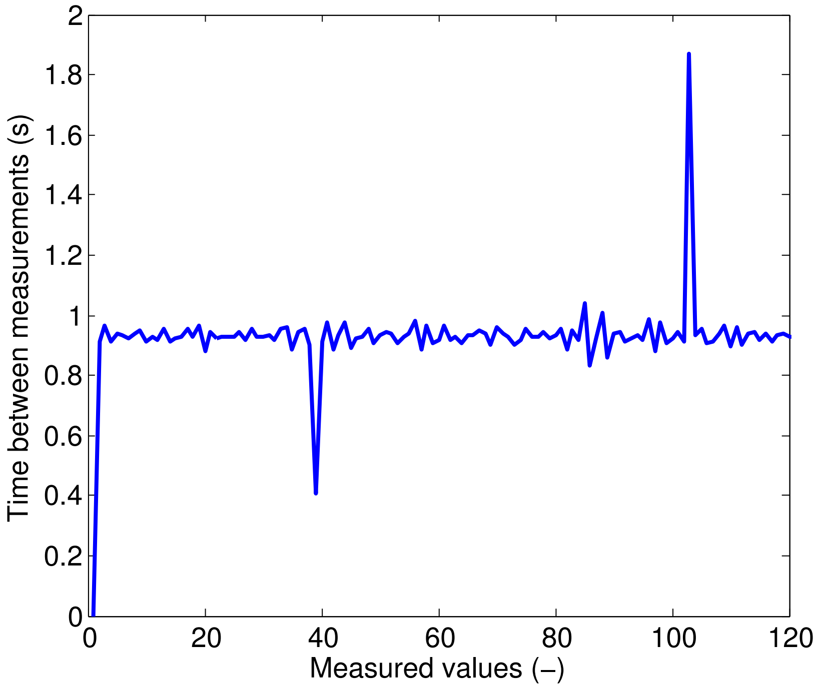

Figure 5.4: Time between measurements of receiver 91, transmitter placed in a position 2

5.2 Creating data vectors periodically from values in a chosen time interval

The new method of creating datasets also computes average and median values from measured signal qualities. The difference is, that these average and median values are determined from all measured signal qualities in a chosen time interval, not from a fixed number of them. For example, the notation $\overline{\mathrm{D}}_{2s}$ shows, that this dataset contains averaged values and the time interval is two seconds. This approach allows us to create a new data vector $\mathbf{Q}$ regularly, independently on measured values. For the following comparison, we used this approach to fill datasets with data vectors $\mathbf{Q}$. These data vectors were created once in a second, and the time intervals were from two to ten seconds for training datasets and from two to twenty seconds for testing datasets. If a certain receiver hadn't measured any value in a chosen time interval, we supposed, that a signal quality at this receiver equals zero. The results are in Table 5.6, Table 5.7, Table 5.8 and Table 5.9.

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$
$\overline{\mathrm{D}}_{2s}$	98.60%	93.56%	92.99%	98.04%	97.76%	98.32%
$\overline{\mathrm{D}}_{4s}$	100.00%	95.52%	94.12%	100.00%	100.00%	100.00%
$\overline{\mathrm{D}}_{10s}$	100.00%	96.92%	96.08%	100.00%	100.00%	100.00%
$\overline{\mathrm{D}}_{16s}$	100.00%	98.04%	98.88%	100.00%	100.00%	100.00%
$\overline{\mathrm{D}}_{20s}$	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 5.6: Success rates of J48 and kNN classifiers trained and tested on averaged values

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	95.52%	88.80%	93.00%	98.04%	96.08%	97.48%
$\overline{\mathrm{D}}'_{4s}$	96.64%	87.96%	94.12%	100.00%	99.44%	99.72%
$\overline{\mathrm{D}}'_{10s}$	98.32%	86.83%	96.07%	100.00%	100.00%	100.00%
$\overline{\mathrm{D}}'_{16s}$	100.00%	90.20%	98.88%	100.00%	100.00%	100.00%
$\overline{\mathrm{D}}'_{20s}$	100.00%	89.08%	100.00%	100.00%	100.00%	100.00%

Table 5.7: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

As shown in the tables above, classifiers trained on the datasets created with the new algorithm in many cases managed to classify all instances correctly. We will not compare this method of creating data vectors **Q** to the previous method, due to the different units.

In the following tests, we exchanged datasets used for training with datasets used for testing. In the earlier tests, we used the more diverse datasets to train our classifiers, and the classifiers were tested on the less diverse datasets. The goal of this test is to determine the importance of the diversity of values used for training classifiers.

As we can see from the results in Table 5.10, Table 5.11, Table 5.12 and Table 5.13, training classifiers on datasets with less diverse values caused

	J48			kNN		
tt \ tn	$\tilde{D}_{2s}$	$\tilde{D}_{4s}$	$\tilde{D}_{10s}$	$\tilde{D}_{2s}$	$\tilde{D}_{4s}$	$\tilde{D}_{10s}$
$\tilde{D}_{2s}$	98.32%	97.20%	95.80%	98.04%	98.88%	99.44%
$\tilde{D}_{4s}$	99.16%	98.32%	95.80%	100.00%	100.00%	100.00%
$\tilde{D}_{10s}$	100.00%	100.00%	97.48%	100.00%	100.00%	100.00%
$\tilde{D}_{16s}$	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
$\tilde{D}_{20s}$	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 5.8: Success rates of J48 and kNN classifiers trained and tested on median values

	J48			kNN		
tt \ tn	$\tilde{D}'_{2s}$	$\tilde{D}'_{4s}$	$\tilde{D}'_{10s}$	$\tilde{D}'_{2s}$	$\tilde{D}'_{4s}$	$\tilde{D}'_{10s}$
$\tilde{D}'_{2s}$	95.24%	96.08%	95.80%	98.04%	97.20%	96.92%
$\tilde{D}'_{4s}$	95.52%	98.04%	95.80%	99.44%	99.44%	99.72%
$\tilde{D}'_{10s}$	97.76%	100.00%	97.48%	100.00%	100.00%	100.00%
$\tilde{D}'_{16s}$	97.48%	100.00%	100.00%	100.00%	100.00%	100.00%
$\tilde{D}'_{20s}$	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%

Table 5.9: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

	J48			kNN		
tt \ tn	$\overline{D}_{2s}$	$\overline{D}_{4s}$	$\overline{D}_{10s}$	$\overline{D}_{2s}$	$\overline{D}_{4s}$	$\overline{D}_{10s}$
$\overline{D}_{2s}$	87.11%	83.19%	84.87%	91.60%	90.20%	89.92%
$\overline{D}_{4s}$	90.48%	87.68%	88.80%	94.17%	93.00%	93.84%
$\overline{D}_{10s}$	94.68%	94.40%	94.12%	98.32%	98.60%	97.48%
$\overline{D}_{16s}$	96.36%	96.08%	97.76%	98.88%	98.60%	100.00%
$\overline{D}_{20s}$	96.08%	96.64%	99.16%	99.72%	100.00%	100.00%

Table 5.10: Success rates of J48 and kNN classifiers trained and tested on averaged values (exchanged datasets)

decrease in a correctly classified instances.

In this section, localization experiments were performed. For the simple case with three receivers, we managed to determine a room, in which a person was located, using classification algorithms. The next step is to apply our

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	87.11%	83.19%	84.87%	90.48%	90.76%	88.52%
$\overline{\mathrm{D}}'_{4s}$	90.48%	87.68%	88.80%	91.04%	91.60%	93.28%
$\overline{\mathrm{D}}'_{10s}$	94.68%	94.40%	94.12%	93.28%	94.96%	96.36%
$\overline{\mathrm{D}}'_{16s}$	96.36%	96.08%	97.76%	89.92%	93.84%	95.24%
$\overline{\mathrm{D}}'_{20s}$	96.08%	96.64%	99.16%	88.24%	94.12%	94.68%

Table 5.11: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations (exchanged datasets)

	J48			kNN		
tt \ tn	$\tilde{\mathrm{D}}_{2s}$	$\tilde{\mathrm{D}}_{4s}$	$\tilde{\mathrm{D}}_{10s}$	$\tilde{\mathrm{D}}_{2s}$	$\tilde{\mathrm{D}}_{4s}$	$\tilde{\mathrm{D}}_{10s}$
$\tilde{\mathrm{D}}_{2s}$	88.52%	82.35%	74.51%	90.48%	91.88%	89.64%
$\tilde{\mathrm{D}}_{4s}$	91.60%	86.83%	77.87%	94.40%	92.72%	93.56%
$\tilde{\mathrm{D}}_{10s}$	95.52%	91.88%	83.19%	97.20%	98.60%	98.60%
$\tilde{\mathrm{D}}_{16s}$	95.52%	92.16%	86.55%	98.32%	99.72%	100.00%
$\tilde{\mathrm{D}}_{20s}$	96.36%	95.52%	88.24%	99.44%	100.00%	100.00%

Table 5.12: Success rates of J48 and kNN classifiers trained and tested on median values (exchanged datasets)

	J48			kNN		
tt \ tn	$\tilde{\mathrm{D}}'_{2s}$	$\tilde{\mathrm{D}}'_{4s}$	$\tilde{\mathrm{D}}'_{10s}$	$\tilde{\mathrm{D}}'_{2s}$	$\tilde{\mathrm{D}}'_{4s}$	$\tilde{\mathrm{D}}'_{10s}$
$\tilde{\mathrm{D}}'_{2s}$	88.52%	82.35%	74.51%	90.48%	90.76%	88.24%
$\tilde{\mathrm{D}}'_{4s}$	91.60%	86.83%	77.87%	91.88%	91.60%	92.72%
$\tilde{\mathrm{D}}'_{10s}$	95.52%	91.88%	83.19%	91.32%	93.56%	95.79%
$\tilde{\mathrm{D}}'_{16s}$	95.52%	92.16%	86.55%	85.99%	93.28%	95.24%
$\tilde{\mathrm{D}}'_{20s}$	96.36%	95.52%	88.24%	88.52%	94.96%	95.52%

Table 5.13: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations (exchanged datasets)

findings on a more complex scenario with six receivers.

5.3 Experiments with six receivers

In this section we aim to examine capabilities of a six-receiver network for the task of localization.

Fig. 5.5 and Fig. 5.6 show areas, where following experiments were carried out. In the first experiment, the person with transmitters tried to cover entire areas to get diverse values, similarly to the experiment with three receivers. This wasn't possible in the second experiment, so values were measured from one place in each room.

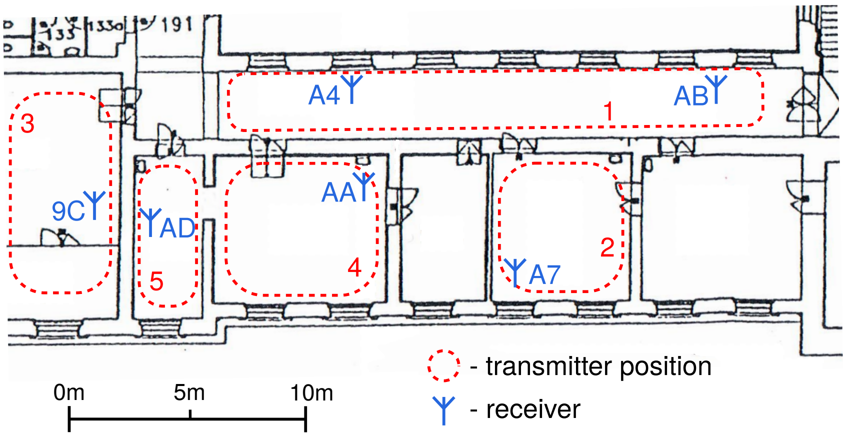

Figure 5.5: Setting for the first experiment with six receivers

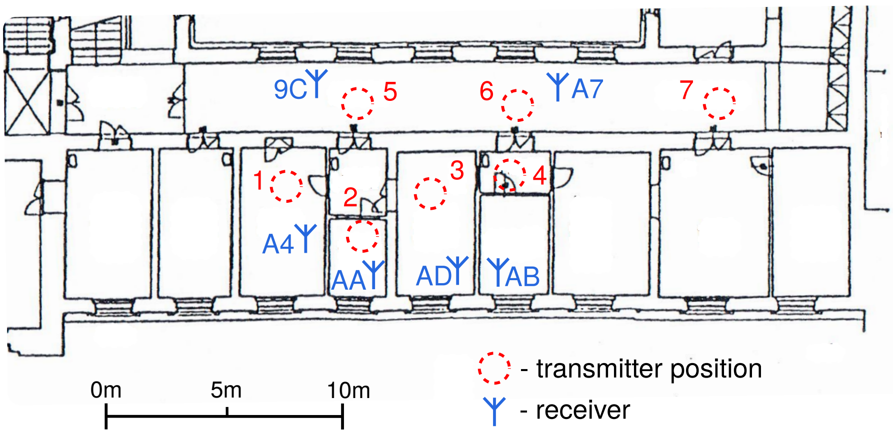

Figure 5.6: Setting for the second experiment with six receivers

Datasets for these experiments were prepared similarly to the datasets used in previous experiments with three receivers. We applied both methods of creating data vectors $\mathbf{Q}$; the first one, using a fixed number of measured signal qualities to create $\mathbf{Q}$; and the second one, creating $\mathbf{Q}$ periodically, from all values in a chosen time interval. Furthermore, we added a new method of training classifiers. In our experiments, we use a low number of receivers. To classify a data vector $\mathbf{Q}$, the classifiers are using values measured by each receiver in a network. In a hospital, however, a larger network of receivers could be used. A potential malfunction of one or more of these receivers could greatly influence a localization process. For this reason, we designed a new method of training classifiers, which could use only a subset from all receivers in a network.

5.3.1 Training classifiers on combinations of receivers

The principle of this method is, that we find all k-combinations from the all n receivers. After that, we will train a classifier on each one of these combinations. This will create $\binom{n}{k}$ separate classifiers. Then, to classify a data vector $\mathbf{Q}$, we can choose a classifier, which will use k values from this vector to determine it's class. We can use different methods for selecting a classifier, on which the classification will be based. For example, we can use the k receivers, which measured the biggest signal quality values. In this work, however, we will not examine methods of selecting single classifiers. We will classify data vectors $\mathbf{Q}$ by the majority of the $\binom{n}{k}$ classifiers.

5.3.2 First setting

Datasets used in the following experiment were created from the values measured in an environment shown in Fig. 5.5.

For this experiment, we used the first method to create datasets. The success rates of classifiers trained and tested on these datasets are in the following tables.

From the tables Table 5.14, Table 5.15, Table 5.16, Table 5.17 and Table 5.18 we can see, that classifiers trained on these datasets haven't achieved as high success rates as in the experiments with three receivers. The decrease was less distinct in case of kNN classifier than with J48 classifier, however, the 100% success rates weren't achieved.

Thereafter we used the second method to created datasets. The data vectors $\mathbf{Q} \in D_m$ were created in the same way as in the experiment with three receivers.

tt \ tn	D_{upd}	$\overline{D}_5$	$\overline{D}_9$	$\overline{D}_{19}$
D_{upd}	67.23%	69.50%	69.84%	58.05%
D_5	76.02%	65.50%	66.66%	56.73%
D_9	76.92%	64.84%	70.33%	57.14%
D_{19}	80.00%	70.00%	72.50%	60.00%

Table 5.14: Success rates of J48 classifier trained and tested on averaged values

tt \ tn	D_{upd}	$\overline{D}_5$	$\overline{D}_9$	$\overline{D}_{19}$
D_{upd}	84.24%	82.99%	86.17%	87.87%
D_5	87.72%	87.72%	89.47%	91.81%
D_9	86.81%	89.01%	92.31%	94.51%
D_{19}	90.00%	95.00%	95.00%	95.00%

Table 5.15: Success rates of kNN classifier trained and tested on averaged values

	J48			kNN		
tt \ tn	$\overline{D}'_5$	$\overline{D}'_9$	$\overline{D}'_{19}$	$\overline{D}'_5$	$\overline{D}'_9$	$\overline{D}'_{19}$
$\overline{D}'_5$	62.57%	61.99%	50.88%	90.64%	90.06%	83.63%
$\overline{D}'_9$	67.03%	68.13%	53.85%	94.51%	94.51%	87.91%
$\overline{D}'_{19}$	60.00%	60.00%	60.00%	95.00%	97.50%	97.50%

Table 5.16: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

	J48			kNN		
tt \ tn	$\tilde{D}_5$	$\tilde{D}_9$	$\tilde{D}_{19}$	$\tilde{D}_5$	$\tilde{D}_9$	$\tilde{D}_{19}$
$\tilde{D}_5$	62.57%	54.38%	61.99%	84.21%	85.38%	89.47%
$\tilde{D}_9$	67.03%	57.14%	61.54%	83.52%	85.71%	87.91%
$\tilde{D}_{19}$	67.50%	57.50%	62.50%	90.00%	87.50%	95.00%

Table 5.17: Success rates of J48 and kNN classifiers trained and tested on median values

In tables Table 5.19, Table 5.20, Table 5.21 and Table 5.22 we can see, that

	J48			kNN		
tt \ tn	$\tilde{\mathrm{D}}'_5$	$\tilde{\mathrm{D}}'_9$	$\tilde{\mathrm{D}}'_{19}$	$\tilde{\mathrm{D}}'_5$	$\tilde{\mathrm{D}}'_9$	$\tilde{\mathrm{D}}'_{19}$
$\tilde{\mathrm{D}}'_5$	59.65%	43.27%	53.22%	87.13%	90.06%	84.80%
$\tilde{\mathrm{D}}'_9$	64.84%	45.05%	56.04%	91.21%	92.31%	86.81%
$\tilde{\mathrm{D}}'_{19}$	60.00%	47.50%	60.00%	97.50%	97.50%	95.00%

Table 5.18: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$
$\overline{\mathrm{D}}_{2s}$	95.24%	96.08%	95.80%	81.79%	81.24%	80.48%
$\overline{\mathrm{D}}_{4s}$	95.52%	98.04%	95.80%	83.64%	82.88%	82.88%
$\overline{\mathrm{D}}_{10s}$	97.76%	100.00%	97.48%	87.68%	85.93%	86.04%
$\overline{\mathrm{D}}_{16s}$	97.48%	100.00%	100.00%	91.49%	89.53%	88.77%
$\overline{\mathrm{D}}_{20s}$	100.00%	100.00%	100.00%	94.55%	92.37%	91.60%

Table 5.19: Success rates of J48 and kNN classifiers trained and tested on averaged values

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	83.53%	63.69%	82.12%	84.84%	85.17%	83.97%
$\overline{\mathrm{D}}'_{4s}$	83.97%	74.05%	85.06%	87.79%	87.78%	86.04%
$\overline{\mathrm{D}}'_{10s}$	84.30%	80.26%	84.84%	92.48%	92.26%	88.55%
$\overline{\mathrm{D}}'_{16s}$	82.77%	83.21%	85.39%	91.60%	92.37%	88.11%
$\overline{\mathrm{D}}'_{20s}$	82.88%	84.73%	86.37%	93.35%	94.55%	89.42%

Table 5.20: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

J48 classifier managed to classify 100% instances correctly, if the datasets contained averaged values.

In the following tables, a success rates of classification based on method described in Subsection 5.3.1 are shown. The number of receivers, on which the classification is based is $k = 3$.

Tables Table 5.23, Table 5.24, Table 5.25 and Table 5.26 show success

tt \ tn	J48 $\tilde{D}_{2s}$	J48 $\tilde{D}_{4s}$	J48 $\tilde{D}_{10s}$	kNN $\tilde{D}_{2s}$	kNN $\tilde{D}_{4s}$	kNN $\tilde{D}_{10s}$
$\tilde{D}_{2s}$	82.99%	71.21%	82.12%	82.22%	80.37%	80.37%
$\tilde{D}_{4s}$	83.64%	74.92%	85.06%	83.53%	80.92%	81.13%
$\tilde{D}_{10s}$	83.97%	78.19%	84.84%	85.60%	80.92%	83.10%
$\tilde{D}_{16s}$	82.55%	80.37%	85.39%	88.33%	83.75%	85.61%
$\tilde{D}_{20s}$	83.21%	81.90%	86.37%	89.86%	85.17%	87.90%

Table 5.21: Success rates of J48 and kNN classifiers trained and tested on median values

tt \ tn	J48 $\tilde{D}'_{2s}$	J48 $\tilde{D}'_{4s}$	J48 $\tilde{D}'_{10s}$	kNN $\tilde{D}'_{2s}$	kNN $\tilde{D}'_{4s}$	kNN $\tilde{D}'_{10s}$
$\tilde{D}'_{2s}$	83.42%	66.41%	75.46%	85.93%	86.04%	84.30%
$\tilde{D}'_{4s}$	84.08%	72.74%	80.04%	88.44%	88.11%	85.28%
$\tilde{D}'_{10s}$	84.41%	76.66%	81.03%	91.60%	91.16%	88.11%
$\tilde{D}'_{16s}$	85.17%	78.63%	82.33%	92.04%	92.14%	87.02%
$\tilde{D}'_{20s}$	85.93%	80.48%	81.90%	93.57%	94.77%	87.79%

Table 5.22: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

tt \ tn	J48 $\overline{D}_{2s}$	J48 $\overline{D}_{4s}$	J48 $\overline{D}_{10s}$	kNN $\overline{D}_{2s}$	kNN $\overline{D}_{4s}$	kNN $\overline{D}_{10s}$
$\overline{D}_{2s}$	74.26%	77.86%	71.86%	76.99%	77.64%	76.44%
$\overline{D}_{4s}$	76.44%	80.58%	74.15%	80.37%	80.04%	79.60%
$\overline{D}_{10s}$	76.88%	81.57%	73.82%	81.35%	81.90%	79.49%
$\overline{D}_{16s}$	76.77%	80.26%	72.19%	80.91%	81.24%	79.60%
$\overline{D}_{20s}$	75.57%	79.71%	71.97%	81.78%	81.78%	80.37%

Table 5.23: Success rates of J48 and kNN classifiers trained and tested on averaged values

rates of classification based on a method described in Subsection 5.3.1. As we can see, the success rates were lower, than in a case of classification based on all receivers. This approach, however, could be potentially improved, if combined with an appropriate method of selecting receivers.

tt \ tn	J48 $\overline{\mathrm{D}}'_{2s}$	J48 $\overline{\mathrm{D}}'_{4s}$	J48 $\overline{\mathrm{D}}'_{10s}$	kNN $\overline{\mathrm{D}}'_{2s}$	kNN $\overline{\mathrm{D}}'_{4s}$	kNN $\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	71.43%	69.68%	64.23%	72.41%	71.21%	70.88%
$\overline{\mathrm{D}}'_{4s}$	74.81%	76.88%	69.90%	77.21%	77.10%	75.25%
$\overline{\mathrm{D}}'_{10s}$	75.46%	83.10%	74.81%	78.95%	80.92%	78.84%
$\overline{\mathrm{D}}'_{16s}$	72.74%	80.70%	74.70%	76.66%	77.64%	80.59%
$\overline{\mathrm{D}}'_{20s}$	71.65%	77.97%	74.92%	74.70%	77.97%	80.59%

Table 5.24: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

tt \ tn	J48 $\tilde{\mathrm{D}}_{2s}$	J48 $\tilde{\mathrm{D}}_{4s}$	J48 $\tilde{\mathrm{D}}_{10s}$	kNN $\tilde{\mathrm{D}}_{2s}$	kNN $\tilde{\mathrm{D}}_{4s}$	kNN $\tilde{\mathrm{D}}_{10s}$
$\tilde{\mathrm{D}}_{2s}$	74.26%	77.86%	71.86%	78.95%	77.64%	76.44%
$\tilde{\mathrm{D}}_{4s}$	76.44%	80.58%	74.15%	80.04%	79.60%	78.62%
$\tilde{\mathrm{D}}_{10s}$	76.88%	81.57%	73.83%	80.26%	80.37%	79.34%
$\tilde{\mathrm{D}}_{16s}$	76.77%	80.26%	72.19%	80.81%	79.83%	79.28%
$\tilde{\mathrm{D}}_{20s}$	75.57%	79.72%	71.97%	80.15%	80.37%	81.13%

Table 5.25: Success rates of J48 and kNN classifiers trained and tested on median values

tt \ tn	J48 $\tilde{\mathrm{D}}'_{2s}$	J48 $\tilde{\mathrm{D}}'_{4s}$	J48 $\tilde{\mathrm{D}}'_{10s}$	kNN $\tilde{\mathrm{D}}'_{2s}$	kNN $\tilde{\mathrm{D}}'_{4s}$	kNN $\tilde{\mathrm{D}}'_{10s}$
$\tilde{\mathrm{D}}'_{2s}$	70.23%	69.79%	64.12%	73.28%	71.76%	70.99%
$\tilde{\mathrm{D}}'_{4s}$	74.15%	74.81%	67.94%	77.54%	75.90%	74.26%
$\tilde{\mathrm{D}}'_{10s}$	76.34%	79.93%	72.85%	78.95%	78.74%	78.30%
$\tilde{\mathrm{D}}'_{16s}$	74.15%	79.28%	71.86%	76.44%	76.99%	78.52%
$\tilde{\mathrm{D}}'_{20s}$	72.41%	78.08%	70.67%	75.13%	76.34%	78.52%

Table 5.26: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

5.3.3 Second setting

Datasets for the following experiments were created from values measured in an environment shown in Fig. 5.6. As we previously said, these values were measured from a one place in each room.

We performed classification tests using classifiers trained on datasets, which were created as described in Section 5.2 and Subsection 5.3.1. The results of the first experiment are in tables Table 1, Table 2, Table 3 and Table 4. The results of the second experiment are in tables Table 5, Table 6, Table 7 and Table 8. These tables are placed in Appendix. As we can see from the achieved success rates, it is important to prepare datasets from a values measured in a whole room and not from a one spot.

5.4 Localization of a dynamic object

The purpose of this experiment is to investigate, how the described algorithms will classify a moving object.

The following experiment was performed in an office environment shown in Fig. 5.7. A person carrying transmitters—$\mathrm{T_A}$ in a hand and $\mathrm{T_B}$ in a pocket—started walking from a point one on the green line. After 41 seconds, he arrived to a point 42. In this experiment, we assume, that he was walking the whole route at a same speed. However, few delays could occur. The data from transmitter $\mathrm{T_A}$ were used to create a training dataset $\overline{\mathrm{D}}_{2s}^{A}$ with a method described in Section 5.2. Then, we used a method described in Subsection 5.3.1 to train classifiers on this dataset, using $k = 3$. The data from the transmitter $\mathrm{T_B}$ were used to create a testing dataset $\overline{\mathrm{D}}_{4s}^{B}$. We used shorter time intervals. In case of long time intervals, if a person is fast, the data vectors would contain values from multiple rooms after processing and it would be harder for classifiers to determine a person's position. In tables Table 5.27 and Table 5.28 are shown individual steps from start to end and a number of classifiers, which classified current step to the same area, using J48 and kNN classifiers.

In the tables we can see, that many times, if a person was standing in an area, the majority of classifiers was classifying him to this area, and if a person was standing between two areas, the same number of classifiers was classifying him to both of these areas. This could be useful for determining a position of a moving person.

	area						area				
step	1	2	3	4	5	step	1	2	3	4	5
1	17	3	0	0	0	22	2	0	1	16	1
2	17	3	0	0	0	23	1	0	3	14	2
3	20	0	0	0	0	24	1	0	1	16	2
4	17	1	0	0	0	25	0	0	1	16	3
5	18	2	0	0	0	26	2	0	2	8	8
6	16	1	0	3	0	27	4	0	4	5	7
7	13	1	0	6	0	28	3	0	1	13	3
8	13	1	0	6	0	29	12	0	1	6	1
9	14	0	0	6	0	30	10	0	0	10	0
10	12	0	0	8	0	31	10	0	0	10	0
11	13	0	0	7	0	32	14	0	0	6	0
12	15	1	0	4	0	33	15	0	0	5	0
13	12	0	0	8	0	34	18	0	0	2	0
14	13	1	0	6	0	35	10	0	4	5	1
15	15	1	0	4	0	36	18	0	0	2	0
16	14	2	0	4	0	37	10	0	4	5	1
17	10	3	1	6	0	38	10	0	4	5	1
18	8	2	1	8	1	39	7	1	0	7	5
19	2	1	6	11	0	40	1	0	0	16	3
20	2	0	6	12	0	41	3	0	12	5	0
21	2	0	1	15	2	42	5	0	12	3	0

Table 5.27: Classification of a dynamic object using J48 algorithm

	area						area				
step	1	2	3	4	5	step	1	2	3	4	5
1	15	5	0	0	0	22	5	1	3	11	0
2	17	3	0	0	0	23	5	2	3	10	0
3	18	1	1	0	0	24	5	2	4	9	0
4	19	1	1	0	0	25	5	2	3	10	0
5	18	2	0	0	0	26	3	0	1	14	2
6	16	4	0	0	0	27	3	0	1	8	8
7	13	2	0	5	0	28	3	0	1	7	9
8	13	2	0	5	0	29	1	0	1	14	4
9	15	0	0	5	0	30	6	0	0	13	1
10	15	0	0	5	0	31	6	0	0	14	0
11	14	0	0	6	0	32	11	0	0	9	0
12	12	0	0	8	0	33	16	0	0	4	0
13	9	0	0	11	0	34	15	2	0	3	0
14	10	0	0	10	0	35	16	1	0	3	0
15	13	1	0	6	0	36	7	0	0	3	10
16	14	3	0	3	0	37	8	0	0	2	10
17	12	7	0	1	0	38	5	2	0	3	10
18	13	5	1	1	0	39	0	0	0	10	10
19	10	5	2	3	0	40	3	0	12	3	2
20	6	2	6	6	0	41	5	0	12	3	0
21	6	2	4	8	0	42	8	0	10	2	0

Table 5.28: Classification of a dynamic object using kNN algorithm

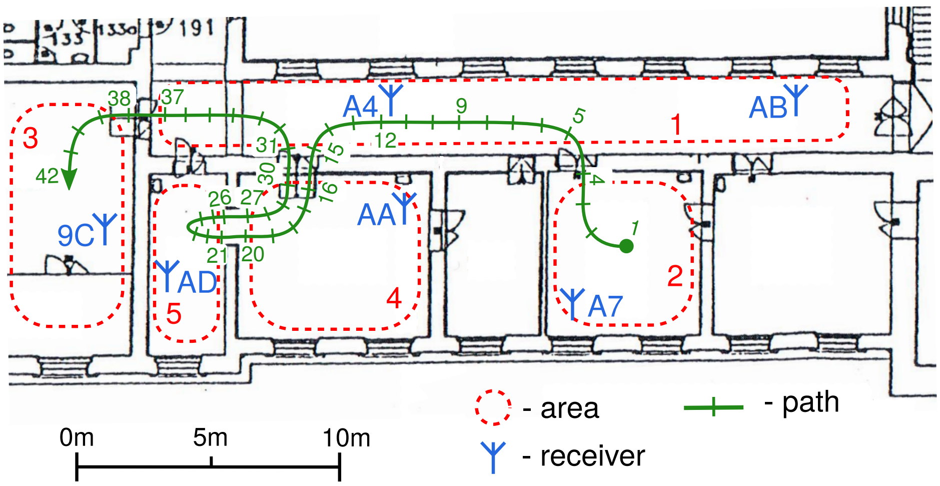

Figure 5.7: A route of a person with transmitter

Chapter 6

Conclusion

The main goal of this work was to investigate, how to localize patients and employees in hospitals using installed network of receivers. Firstly, we examined properties of the transmitters, carried by individuals; and receivers, used for acquiring data from the transmitters. We found out, that this system is useful for determining a person's position, due to the convenient behaviour of the signal qualities measured by a receiver. For example, measured signal qualities were dependent on the distance between the transmitter and the receiver. Another important characteristic of the measured signal quality was, that it was not strongly dependent on the orientation of the transmitter or the receiver.

To determine a person's position from the measured data, we had to employ classification algorithms; we chose the k-nearest neighbours algorithm and C4.5 algorithm. These algorithms are implemented in the Weka system, which is an open source software.

Classification experiments were performed in an office environment, using three and six-receiver systems. Firstly, we measured data in these environments. After that, we created datasets from the measured data in a various ways to determine, which approach suits better for our task. Then we compared selected classifiers trained on these datasets. We found out, that the better way to create datasets from the measured data is to create data vectors periodically, processing measured values from a certain time interval. The size of the interval depends on a current task. If the objects of classification are static, we can process values from a longer time interval, which increases classifiers success rates. If the objects are dynamic, we should use shorter time intervals. We found out, that the best methods of processing values and the best classifier are different for a different experiments, so we can't select one suitable for every occasion. In these experiments we also verified the importance of a the training dataset created from diverse values, measured in a whole rooms.

A method of classification based on a few selected receivers from a set of all receivers was described. This method could be more reliable in a case of receiver malfunctions. Also, we used this method to determine a position of a moving person.

In this work, we performed test on a maximum number of six receivers. To compare classification methods used in this work more thoroughly, further tests with more receivers should be performed.

Bibliography

[1] M. Stella, M. Russo, and D. Begušić. "RF Localization in Indoor Environment". In: *Radioengineering* 21.2 (June 2012).

[2] NXP Laboratories UK Ltd. *Data Sheet: JN5148-001-Myy. JenNet, ZigBee PRO and IEEE802.15.4 Module*. 2010. URL: `http://www.jennic.com/files/product_briefs/JN-DS-JN5148MO-1v4.pdf` (visited on 05/10/2014).

[3] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)". In: *IEEE Std 802.15.4-2003* (2003). DOI: `10.1109/IEEESTD.2003.94389`.

[4] V. Mařík et al. *Umělá inteligence (1)*. 1st ed. Academia, 1993. ISBN: 80-200-0496-3.

[5] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN: 1-55860-238-0.

[6] M. Hall et al. "The WEKA Data Mining Software: An Update". In: *SIGKDD Explorations* 11.1 (2009).

Appendices

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$	$\overline{\mathrm{D}}_{2s}$	$\overline{\mathrm{D}}_{4s}$	$\overline{\mathrm{D}}_{10s}$
$\overline{\mathrm{D}}_{2s}$	39.24%	39.24%	37.22%	41.38%	36.74%	38.76%
$\overline{\mathrm{D}}_{4s}$	46.97%	46.14%	44.11%	48.04%	49.35%	50.54%
$\overline{\mathrm{D}}_{10s}$	52.32%	48.39%	48.28%	48.63%	58.98%	60.05%
$\overline{\mathrm{D}}_{16s}$	83.39%	45.18%	48.99%	47.44%	60.64%	60.76%
$\overline{\mathrm{D}}_{20s}$	53.86%	44.00%	49.70%	47.21%	61.47%	61.24%

Table 1: Success rates of J48 and kNN classifiers trained and tested on averaged values

	J48			kNN		
tt \ tn	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$	$\overline{\mathrm{D}}'_{2s}$	$\overline{\mathrm{D}}'_{4s}$	$\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	41.74%	35.32%	35.20%	41.74%	35.32%	35.20%
$\overline{\mathrm{D}}'_{4s}$	51.01%	50.30%	48.75%	51.01%	50.30%	48.76%
$\overline{\mathrm{D}}'_{10s}$	56.36%	59.69%	64.33%	56.36%	59.69%	64.33%
$\overline{\mathrm{D}}'_{16s}$	52.20%	55.89%	64.45%	52.20%	55.89%	64.45%
$\overline{\mathrm{D}}'_{20s}$	51.13%	53.75%	62.54%	51.13%	53.75%	62.54%

Table 2: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

	J48			kNN		
tt \ tn	$\tilde{\mathrm{D}}_{2s}$	$\tilde{\mathrm{D}}_{4s}$	$\tilde{\mathrm{D}}_{10s}$	$\tilde{\mathrm{D}}_{2s}$	$\tilde{\mathrm{D}}_{4s}$	$\tilde{\mathrm{D}}_{10s}$
$\tilde{\mathrm{D}}_{2s}$	38.53%	39.60%	43.88%	38.41%	34.96%	38.17%
$\tilde{\mathrm{D}}_{4s}$	43.28%	50.42%	53.39%	46.37%	48.04%	50.54%
$\tilde{\mathrm{D}}_{10s}$	43.04%	52.91%	60.04%	48.17%	56.48%	61.36%
$\tilde{\mathrm{D}}_{16s}$	39.59%	51.25%	61.83%	47.44%	58.26%	60.76%
$\tilde{\mathrm{D}}_{20s}$	38.53%	49.94%	62.43%	47.80%	57.91%	60.52%

Table 3: Success rates of J48 and kNN classifiers trained and tested on median values

tt \ tn	J48 $\tilde{\mathrm{D}}'_{2s}$	J48 $\tilde{\mathrm{D}}'_{4s}$	J48 $\tilde{\mathrm{D}}'_{10s}$	kNN $\tilde{\mathrm{D}}'_{2s}$	kNN $\tilde{\mathrm{D}}'_{4s}$	kNN $\tilde{\mathrm{D}}'_{10s}$
$\tilde{\mathrm{D}}'_{2s}$	40.43%	35.32%	37.69%	40.55%	36.39%	35.43%
$\tilde{\mathrm{D}}'_{4s}$	45.78%	45.30%	45.42%	51.49%	51.61%	49.94%
$\tilde{\mathrm{D}}'_{10s}$	48.75%	52.20%	53.86%	55.17%	59.45%	64.09%
$\tilde{\mathrm{D}}'_{16s}$	48.28%	53.27%	57.07%	50.18%	55.53%	61.36%
$\tilde{\mathrm{D}}'_{20s}$	48.87%	53.27%	58.38%	48.99%	53.27%	60.64%

Table 4: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations

tt \ tn	J48 $\overline{\mathrm{D}}_{2s}$	J48 $\overline{\mathrm{D}}_{4s}$	J48 $\overline{\mathrm{D}}_{10s}$	kNN $\overline{\mathrm{D}}_{2s}$	kNN $\overline{\mathrm{D}}_{4s}$	kNN $\overline{\mathrm{D}}_{10s}$
$\overline{\mathrm{D}}_{2s}$	24.49%	25.21%	22.71%	27.94%	24.14%	23.78%
$\overline{\mathrm{D}}_{4s}$	28.41%	32.58%	29.37%	34.13%	32.10%	30.68%
$\overline{\mathrm{D}}_{10s}$	27.59%	37.21%	29.73%	34.60%	35.79%	36.62%
$\overline{\mathrm{D}}_{16s}$	25.45%	34.96%	29.73%	31.87%	33.41%	34.84%
$\overline{\mathrm{D}}_{20s}$	26.63%	34.48%	30.56%	30.68%	32.70%	34.84%

Table 5: Success rates of J48 and kNN classifiers trained and tested on averaged values

tt \ tn	J48 $\overline{\mathrm{D}}'_{2s}$	J48 $\overline{\mathrm{D}}'_{4s}$	J48 $\overline{\mathrm{D}}'_{10s}$	kNN $\overline{\mathrm{D}}'_{2s}$	kNN $\overline{\mathrm{D}}'_{4s}$	kNN $\overline{\mathrm{D}}'_{10s}$
$\overline{\mathrm{D}}'_{2s}$	20.57%	18.55%	17.84%	21.76%	20.57%	20.21%
$\overline{\mathrm{D}}'_{4s}$	27.71%	26.29%	23.54%	30.20%	28.78%	26.04%
$\overline{\mathrm{D}}'_{10s}$	29.96%	33.65%	25.80%	33.77%	33.53%	33.89%
$\overline{\mathrm{D}}'_{16s}$	27.59%	32.46%	25.92%	31.75%	31.03%	32.22%
$\overline{\mathrm{D}}'_{20s}$	27.46%	32.70%	24.85%	31.27%	31.03%	31.51%

Table 6: Success rates of J48 and kNN classifiers trained and tested on averaged values with standard deviations

tn / tt	J48 $\tilde{\mathrm{D}}_{2s}$	J48 $\tilde{\mathrm{D}}_{4s}$	J48 $\tilde{\mathrm{D}}_{10s}$	kNN $\tilde{\mathrm{D}}_{2s}$	kNN $\tilde{\mathrm{D}}_{4s}$	kNN $\tilde{\mathrm{D}}_{10s}$
$\tilde{\mathrm{D}}_{2s}$	25.09%	24.85%	23.31%	28.06%	25.45%	24.13%
$\tilde{\mathrm{D}}_{4s}$	29.37%	30.68%	30.32%	34.96%	31.87%	31.03%
$\tilde{\mathrm{D}}_{10s}$	29.85%	34.24%	35.91%	36.03%	35.43%	34.36%
$\tilde{\mathrm{D}}_{16s}$	30.44%	33.53%	37.21%	34.60%	37.34%	34.00%
$\tilde{\mathrm{D}}_{20s}$	30.92%	33.41%	40.31%	33.77%	35.79%	33.53%

Table 7: Success rates of J48 and kNN classifiers trained and tested on median values

tn / tt	J48 $\tilde{\mathrm{D}}'_{2s}$	J48 $\tilde{\mathrm{D}}'_{4s}$	J48 $\tilde{\mathrm{D}}'_{10s}$	kNN $\tilde{\mathrm{D}}'_{2s}$	kNN $\tilde{\mathrm{D}}'_{4s}$	kNN $\tilde{\mathrm{D}}'_{10s}$
$\tilde{\mathrm{D}}'_{2s}$	20.21%	17.84%	17.00%	23.19%	20.33%	20.33%
$\tilde{\mathrm{D}}'_{4s}$	27.94%	24.85%	21.40%	30.56%	29.13%	25.68%
$\tilde{\mathrm{D}}'_{10s}$	32.46%	29.25%	26.75%	33.53%	33.89%	33.06%
$\tilde{\mathrm{D}}'_{16s}$	31.03%	30.43%	24.85%	33.41%	33.41%	31.39%
$\tilde{\mathrm{D}}'_{20s}$	31.51%	31.27%	24.85%	33.89%	32.94%	31.75%

Table 8: Success rates of J48 and kNN classifiers trained and tested on median values with standard deviations