Masaryk University
Faculty of Informatics

Localization of mobile devices using machine learning

Bachelor's Thesis

Aleš Calábek

Brno, Spring 2020

Declaration

Hereby I declare that this paper is my original authorial work, which I have worked out on my own. All sources, references, and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Aleš Calábek

Advisor: Mgr. Adam Janovský

Acknowledgements

Foremost, I would like to express my gratitude to my advisor Adam Janovský for the advice and guidance he provided throughout the writing of this thesis. I would also like to thank all my friends and family for the moral support.

Abstract

This thesis examines the potential of machine learning for GSM positioning based on received signal strength in urban environments. The goal was to assess two algorithms, namely Support Vector Machine and Multilayer Perceptron, and compare their performance to other localization techniques. The models were evaluated on data collected in the city of Brno, Czech Republic using the scanner application Molotras. I show that modelling signal propagation using the studied algorithms does not provide an advantage over traditional methods due to excessive noise in the data caused by buildings blocking the signal's path. Finally, I propose an alternative approach to using neural networks that works on the principle of fingerprinting.

Keywords

localization, GSM, RSSI, machine learning, Support Vector Machine, Multilayer Perceptron

Contents

1 Introduction **1**

2 Radio technology **3**
2.1 *Wireless Communication* 3
2.2 *GSM Network* 4
2.3 *Localization* 6
2.3.1 Time of Arrival 6
2.3.2 Received signal strength 7
2.3.3 Angle of Arrival 7

3 Machine learning **11**
3.1 *Machine learning* 11
3.2 *Support Vector Machine* 12
3.2.1 Support Vector Regression 13
3.3 *Artificial Neural Networks* 14
3.3.1 Perceptron 14
3.3.2 Multilayer Perceptron 16

4 Dataset **19**
4.1 *Used technologies* 19
4.2 *Data collection* 20
4.3 *Data processing* 22

5 Evaluation **27**
5.1 *Training* 27
5.2 *Results* 29
5.3 *Comparison* 31

6 Conclusion **35**

A Evaluation **43**

B Archive structure **45**

List of Tables

4.1 Number of received signal strength records per one measurement. *23*

5.1 Fingerprinting in the city centre showed the lowest localization error as compared to other positioning methods that otherwise performed comparably. Data from [1]. *32*

5.2 ML models showed higher localization error than the traditional methods when tested on the village dataset. Data from [1]. *32*

A.1 When trained on data from the city centre, both SVR and MLP showed similar localization error to the baseline model. *43*

A.2 Training on data from the residential areas significantly impaired the SVR's performance in the city centre. *43*

A.3 The implemented models performed worse when working with only one signal value at a time. Table shows positioning error when evaluated in the city centre. *44*

List of Figures

2.1 Location of the receiver can be found at the intersection of imaginary spheres formed around each satellite [11]. *7*

2.2 Two angles to known locations are sufficient to locate a mobile device using triangulation [13]. *8*

3.1 Multiple hyperplanes can consistently divide an input space (left). The optimal hyperplane maximizes the margins between it and the samples on either side (right). Images from [16]. *12*

3.2 The classes cannot always be separated directly (left). Kernels project the data into a higher dimension where it may become possible to find a dividing hyperplane (right). Images from [21]. *13*

3.3 Different activation functions are suitable for different tasks. They also greatly influence prediction accuracy and computation time. *15*

3.4 Neurons in one layer of the MLP connect to all neurons in the subsequent layer [25]. *16*

3.5 During the gradient descent, weights are adjusted in the direction that reduces error [26]. *17*

4.1 I have split the collected measurements into two datasets: city centre (red) and residential (blue). Includes data from [1]. *21*

4.2 There does not seem to be a clear relationship between received signal strength and distance. *25*

5.1 The MLP showed identical localization accuracy to the baseline model. *29*

5.2 All models showed worse performance when working with only one signal value at a time instead of using data from all available cells at once. *30*

5.3 The positioning error in the village increased by an order of magnitude. *30*

1 Introduction

Mobile phones are a part of everyday life for most people in the developed world. They provide a variety of convenient utilities, among others, the location-based services used in mobile navigation and fitness applications. The majority of people, however, does not realize that their location can be tracked even with their phone's location services turned off.

One way of locating devices is to take advantage of its connection to the cellular network. As signal strength readings include the identity of the transmitting cell towers, the phone can be located based on its distance to the towers' positions.

In this thesis, I examined the potential of using machine learning (ML) for improving the localization accuracy. I studied two ML models, namely Support Vector Machine and Multilayer Perceptron. The implementation was done in the Python programming language and used the scikit-learn and Keras libraries. Both models were evaluated on data I have collected in the city centre of Brno using the *Molotras* scanner application [1]. Finally, I compared my results with traditional positioning techniques.

The structure of this thesis is as follows: In Chapter 2, I introduce the fundamentals of radio signal propagation, the structure of a cellular network and several localization approaches. Chapter 3 presents the ML algorithms mentioned above, and Chapter 4 describes the process of collecting, analyzing and processing the data. I elaborate on the specifics of training the models, evaluating their performance and comparing their results in Chapter 5. Finally, Chapter 6 provides a conclusion and suggests possible future work.

2 Radio technology

The primary function of a cellular network is to provide the means for communication. We can, however, use the properties of the transmitted signal itself to find the location of a device in the network.

In the first part of this chapter I introduce the basics of signal propagation. The second part provides information about the structure and workings of the second-generation cellular network. Finally, the third part presents several techniques used for localization.

2.1 Wireless Communication

In modern society, people and devices need to communicate with each other. Common household appliances such as television, radio, Wi-Fi or mobile phones all connect wirelessly. Wireless technology enables us to transmit information over long distances by using electromagnetic (EM) waves without the use of physical point-to-point links.

Frequency, amplitude and phase are the three fundamental characteristics of an EM wave. Information can be encoded into the wave by changing one of the three properties in a process called modulation. Frequency modulation (FM), in particular, is widely known for its use in FM radio broadcasting [2].

The EM spectrum encompasses all EM radiation frequencies ranging from the low-frequency radio waves to the high-frequency gamma rays. However, not all frequencies are suitable for communication purposes.

Signals at the very low end of the spectrum, with wavelengths of thousands of kilometres, can travel halfway around the Earth before they require any form of amplification, but provide very low bandwidth. On the other end of the spectrum, signals can carry an exceptional amount of information but are prone to absorption by rain and reflection off obstacles. For the above reasons, communication technology primarily uses the middle of the EM spectrum—the radio, microwave, infrared, and visible frequencies [3].

Considering the great impact that obstacles have on wave propagation, it is worth elaborating on their effects some more.

When a wave collides with an object, part of its energy is transmitted through, while the rest is reflected back. For instance, reflections may occur when the receiver lies inside a building and the wave must pass through the walls.

Diffraction is another phenomenon affecting the wave, causing it to bend as it travels around an obstacle. This change in direction allows waves to propagate to places behind objects.

Before reaching the receiver, radio waves may reflect and diffract many times. Multiple versions of the signal then arrive at different times and may interfere with each other, making it difficult to decode the resulting signal. Interference also occurs due to other transmissions on the same frequency or the presence of noise. Analyzing these effects is essential in order to understand how waves behave. However, simulating the propagation of radio signals in a real environment would be infeasible because it would require precise knowledge about all objects on the scene [4].

Mathematical models simplify the problem of wave tracing by estimating how much power the signal loses over the path from transmitter to receiver. One of the simplest propagation models is *Free Space Path Loss* which assumes a direct path of the signal with no atmospheric attenuation or interference. The path loss is given by the equation

$$FSPL = \left(\frac{4\pi d}{\lambda}\right)^2, \tag{2.1}$$

where d is the distance from the transmitter to the receiver and λ is the wavelength of the radio signal [5][6][7].

With the knowledge of radio wave properties and their behaviour, we can now better understand how cellular networks use wireless transmission to provide communication services.

2.2 GSM Network

The cellular network forms the foundation of modern communication technology. It enables the transmission of voice, text messages and Internet data all over the world.

The Global System for Mobile Communications (GSM) Network is a second-generation (2G) cellular network. It was first deployed

in 1991 and is still in use today. It comes with many improvements over the first-generation analogue network—notably, higher capacity and digital encryption, allowing for more people to communicate privately.

An essential part of any cellular network is the Base Transceiver Station (BTS). A BTS used in the GSM network consists of the radio transmitter-receiver and its associated antenna. These antennas can be seen on top of masts or tall buildings, where surrounding structures do not block their signal.

Still, high-rise buildings in cities prove themselves to be enough of an obstacle to necessitate higher density of BTS towers to provide coverage of all areas and to handle the increased traffic. Rural regions, on the other hand, are usually covered by fewer but more powerful transmitters placed at high terrain to maximize their range.

It is, however, not possible to prevent overlap of the signals in adjacent cells which may lead to reduced call quality. The unwanted interference can be eliminated by using different frequencies in neighbouring cells and reusing them in cells that are further apart.

Another difficulty that comes with dividing the area into sections is providing a continuous connection when the mobile moves across the boundaries. At all times, the mobile scans available radio channels looking for the most reliable one. When the current serving cell's signal drops below a critical level, the network arranges a *handover*, and the mobile reconnects to a different cell [8][9][10].

During the handover, the mobile and the network need an unambiguous way to refer to the cells. Therefore, every BTS is uniquely distinguishable by four identification numbers:

- Mobile Country Code (MCC) specifies a country,
- Mobile Network Code (MNC) determines the mobile operator,
- Location Area Code (LAC) refers to a group of cells,
- Cell ID (CID) identifies individual BTSs within that group.

With the knowledge of a cell's identity, its position can be easily learned by searching a database. If we could use the properties of the received signal to determine the distance to several known cells, we could find the location of the mobile receiver.

2.3 Localization

The simplest localization method in a cellular network is Cell Identification or Cell ID. In Cell ID, the mobile's and the serving cell's locations are considered to be the same. The accuracy of this method generally ranges in hundreds or thousands of metres and largely depends on the dimensions of the area covered by the cell.

Better accuracies can be achieved by using more elaborate approaches. Time of arrival, received signal strength and angle of arrival are just three of the radio signal properties that enable us to determine the location of the receiver.

2.3.1 Time of Arrival

Time of Arrival (TOA) methods make use of the time it takes a signal to reach the receiver. Distance to the transmitter can be calculated numerically based on the signal's propagation speed and the time delay. With known distances to multiple transmitters, the location of the mobile receiver can be estimated using the method of multilateration.

Multilateration forms imaginary circles around the transmitters' positions with the calculated distances as radii. These circles, in theory, intersect at the precise position of the receiver. In practice, however, the intersecting circles delimit an area, and optimization techniques have to be used to minimize the positioning error.

Arguably the most well-known system based on TOA is the Global Positioning System (GPS). Each satellite in the GPS network broadcasts a report about its position and the time of transmission. With the data from at least four satellites, the mobile receiver can deduce its location in three dimensions, as illustrated in Figure 2.1 [11].

While GPS positioning can provide localization accuracies of tens of centimetres, it comes with several disadvantages. As line of sight to the satellite is a necessity, the use indoors, between high-rise buildings and in mountainous terrain is limited. It is also the most power-consuming method making it infeasible in smaller devices. Moreover, additional hardware in the form a GPS chip is required[12].

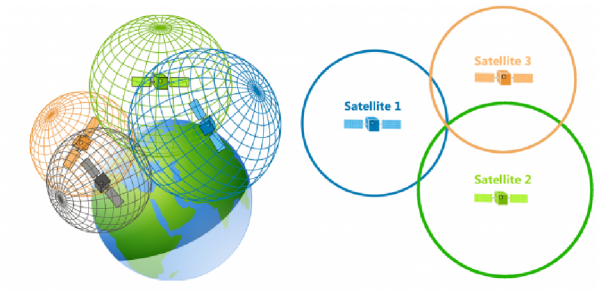


Figure 2.1: Location of the receiver can be found at the intersection of imaginary spheres formed around each satellite [11].

2.3.2 Received signal strength

As radio signal travels through space, it loses energy. With the knowledge of the power radiated by the transmitter P_t and the signal strength P_r received by the mobile device, the path loss PL can be calculated using the equation

$$PL = P_t - P_r. \tag{2.2}$$

A propagation model, such as FSPL mentioned in section 2.1, can be applied to estimate the distance that the signal has travelled. The mobile device then computes its position based on the distance estimates and cell tower locations using multilateration.

Contrary to the TOA methods, techniques based on Received Signal Strength work with no additional equipment and come free with the GSM infrastructure. However, localization accuracy is generally worse due to variations in signal strength caused by interference, multipath propagation and path blocking.

2.3.3 Angle of Arrival

Angle of Arrival (AOA) is the principal component in a radar system. The radar rotates its receiving antenna and notes the changes in received signal strength. The AOA is determined from the direction of the antenna at the time it receives the strongest signal. The radar then uses TOA to calculate the distance to a target and find its location in three dimensions.

When using AOA to locate mobile devices in a cellular network without the aid of TOA, at least two angle measurements are necessary. With two angles to known positions, the mobile station can compute its location by using the method of triangulation, as shown in Figure 2.2.

One advantage of AOA methods is the reduced effect of interference achieved by rejecting signals coming from directions other than those of the transmitter. Still, noise and path blocking limit the localization accuracy in the same way as with the methods based on received signal strength [13].

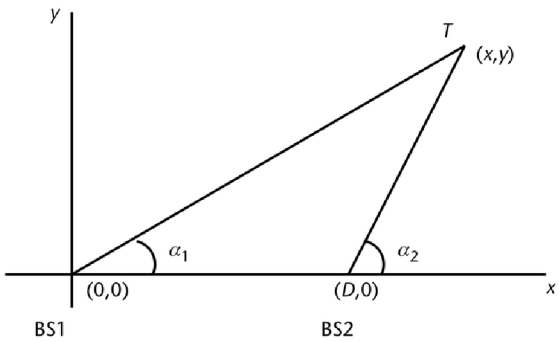


Figure 2.2: Two angles to known locations are sufficient to locate a mobile device using triangulation [13].

Conclusion

Radio waves enable us to communicate wirelessly. We can encode information by modulating their properties such as frequency, amplitude or phase. As radio waves travel through space, they are affected by obstacles in their way. We use mathematical models to estimate the loss of signal power on the wave's path from transmitter to receiver.

The GSM network is a cellular network composed of a grid of BTS towers, each uniquely identifiable. Physical properties of the radio signal can be used to locate devices in the network.

In this thesis, I focus on positioning methods based on received signal strength because they require no additional hardware and can

be readily implemented on the existing infrastructure. I explore the feasibility of improving the traditional techniques by using machine learning algorithms, which are described in the next chapter.

3 Machine learning

Machine learning (ML) systems are part of everyday life. They are used to recommend videos on Youtube or products on Amazon. Google uses ML to provide personalized results in its search engine and to filter spam messages in Gmail's inboxes. Speech recognition and synthesis enables virtual assistants, such as Apple's Siri or Amazon's Alexa, to interact with people using natural language. Machine learning systems can also drive cars, unlock phones with facial recognition, detect credit card frauds, and much more.

In the first part of this chapter, I introduce the concept of machine learning and describe one class of ML algorithms. Parts two and three then provide detailed information about the two algorithms used in this thesis, namely the Support Vector Machine and Multilayer Perceptron.

3.1 Machine learning

Machine learning enables computers to perform tasks commonly done by humans. It provides machines with the ability to learn and improve from experience without being explicitly programmed to solve a problem. As a field of study, ML is a part of a broader discipline of Artificial Intelligence [14].

ML methods can be divided into several categories based on the type of data they process and the type of problem they intend to solve. In this thesis, I concentrate on algorithms that fall into the *supervised learning* category.

Supervised learning is typically used for classification or regression problems. During classification, we want to decide to which class an object belongs. For instance, deciding whether or not an email is spam is a classification problem. With regression, we want to predict a value in a continuous range, such as the stock price in a stock market.

In both classification and regression, the goal is to find the connection between the inputs and the outputs. A learning algorithm processes the *training data* and creates a statistical model capturing this relationship. The quality of the model heavily depends on the

quality of the dataset and noisy data with incorrect or missing values will dramatically reduce the model's accuracy.

Another aspect concerning performance is *overfitting*. Overfitting refers to creating a model that fits the training data well and can predict outputs accurately, but is not able to generalize to data it has not seen before [15].

Some of the widely adopted supervised learning algorithms include the Support Vector Machine and Multilayer Perceptron, which I used in this thesis to locate devices in the GSM network, and which are described in the next two sections.

3.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised learning algorithm typically used for classification. It looks at the input data as points in space and tries to divide the space so that instances of different categories are separated. New data points are then classified based on their position relative to the decision boundary, or the *hyperplane*.

A hyperplane is a subspace of a dimension one less than that of its ambient space. For example, in two dimensions, the hyperplane is a line.

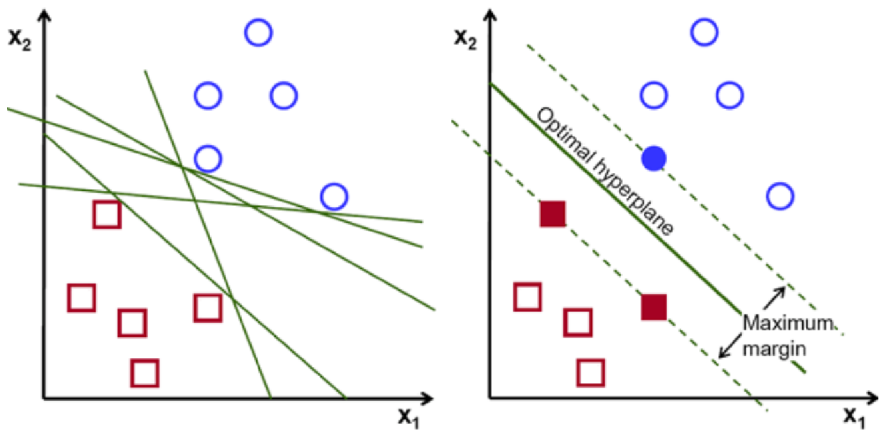


Figure 3.1: Multiple hyperplanes can consistently divide an input space (left). The optimal hyperplane maximizes the margins between it and the samples on either side (right). Images from [16].

Figure 3.1 demonstrates that multiple hyperplanes can consistently divide an input space. The further a data point is from the decision boundary, the more confident we can be about predicting its class. Therefore, we want to find the hyperplane with maximum *margins* between it and the samples on both sides. The data points closest to the hyperplane are then called the *support vectors*.

Nevertheless, a hyperplane cannot always divide the input data directly, as illustrated by Figure 3.2. The SVM overcomes this problem by using *soft margins* and *kernels*. Soft margins tolerate the misclassification of some data points and the SVM tries to balance the trade-off between maximizing the margins and minimizing the mistakes. A kernel is a function that transforms the input space into a higher dimension and enables the SVM to model non-linear relationships [17][18].

3.2.1 Support Vector Regression

The SVM can also be extended to handle regression problems. Instead of separating the dataset, Support Vector Regression (SVR) tries to find a hyperplane that fits the data and includes all data points within its margins. The width of the margins then indicates how much deviation the models allows [19][20].

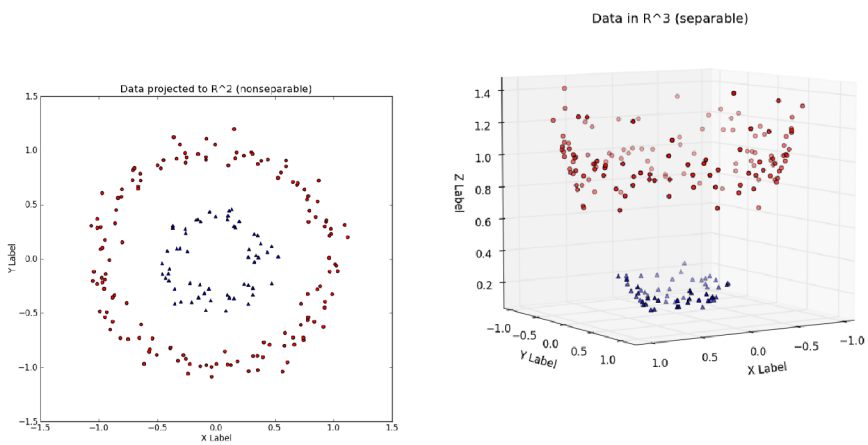


Figure 3.2: The classes cannot always be separated directly (left). Kernels project the data into a higher dimension where it may become possible to find a dividing hyperplane (right). Images from [21].

3.3 Artificial Neural Networks

The idea of an Artificial Neural Network (ANN) was first introduced in the 1950s and was inspired by the workings of a human brain. As in the brain, the output of an ANN depends on the connections between individual processing units, or *neurons*. However, contrary to our brains that work massively in parallel, computers perform their operations sequentially and cannot simulate large networks in real-time. Still, ANNs are a powerful class of algorithms capable of solving many challenging problems, some of which I mentioned at the beginning of this chapter.

3.3.1 Perceptron

The perceptron is a mathematical model representing a single neuron. It accepts real number values x_i as inputs and calculates their weighted sum ξ (also called the *inner potential*) using corresponding weights w_i.

$$\xi = \sum_{i=0}^{N} w_i x_i \tag{3.1}$$

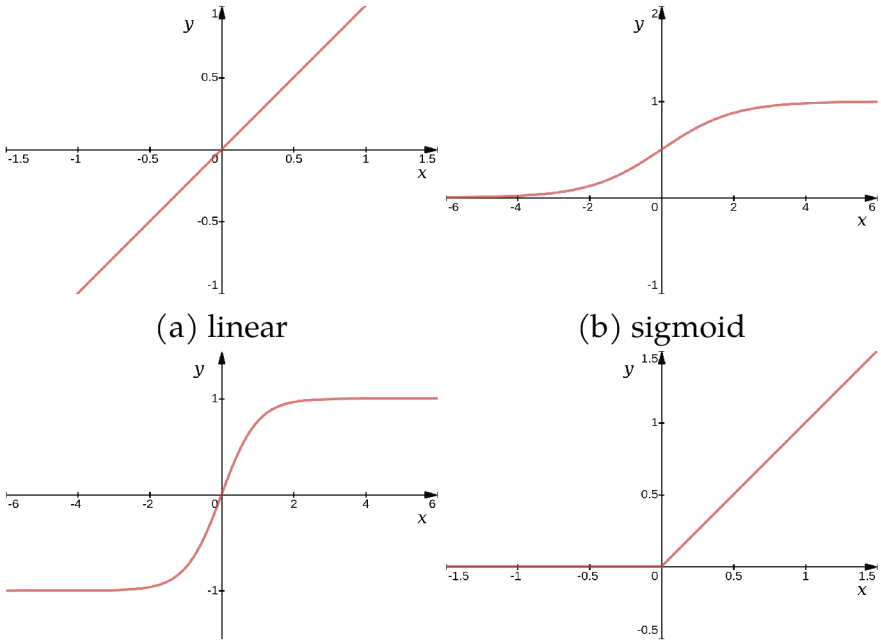
The output value of the perceptron y is then obtained by applying the *activation function* f to the inner potential.

$$y = f(\xi) \tag{3.2}$$

Activation functions play a critical role in the learning process as different functions are suitable for different tasks. They can have a significant influence on prediction accuracy as well as the computation time. Figure 3.3 shows some of the most commonly used activation functions that include the *linear, sigmoid, tanh* or *ReLU* functions.

The linear function of the form $f(x) = cx$ is one of the simplest activation functions and is fast to compute. The downside is that it cannot express non-linear relationships because the combination of linear functions is again a linear function. It is usually used in the last layer of the network to predict unbounded values.

The sigmoid function $f(x) = \frac{1}{1+e^{-}x}$ maps the outputs to a range between 0 and 1, which is especially useful for predicting probabilities. However, for very high or low values of x, there is almost no change



(a) linear (b) sigmoid

(c) tanh (d) ReLU

Figure 3.3: Different activation functions are suitable for different tasks. They also greatly influence prediction accuracy and computation time.

to the predictions which may lead to the *vanishing gradient* problem that causes longer training times of the network.

The hyperbolic tangent or tanh $f(x) = \frac{2}{1+e^{-2x}} - 1$ is similar to the sigmoid function. It ranges from -1 to 1, meaning the negative inputs are mapped to negative values, and the zero inputs are mapped near zero. Along with the sigmoid, tanh is expensive to compute, and suffers from the vanishing gradient problem as well.

The Rectified Linear Unit or ReLU $f(x) = max(0, x)$ is another non-linear activation function. At the negative end of its range, it is also affected by vanishing gradient. Still, it is ideal for use in larger networks due to its computational efficiency making it possibly the most used activation function today [22][23][24].

3.3.2 Multilayer Perceptron

Individual perceptrons can be organized into a structure of interconnected layers to form one of the simplest neural network models—the Multilayer Perceptron. The combination of non-linear activations and the network arrangement makes the MLP capable of approximating an arbitrary function.

The layers of an MLP are of three types. The first layer accepts the input data and forwards them into the network, hence the *input layer*. It is followed by one or more *hidden layers* that make up the core of the network. Perceptrons in the hidden layers use various activation functions to capture the essence of the data and generalize the relations between the inputs. In practice, there is no general best number of hidden layers, and it has to be tailored to the task at hand. Finally, the *output layer* is the result of the MLP and is interpreted according to the specific problem being solved.

As shown in Figure 3.4, every neuron in one layer connects to all neurons in the previous layer. The *backpropagation* algorithm then adjusts the weights associated with these connections, and that is how the network learns.

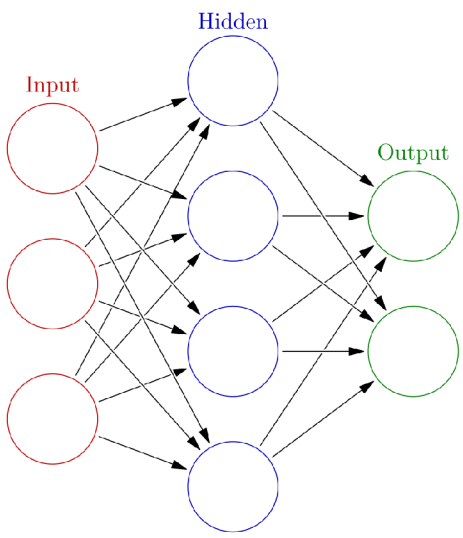


Figure 3.4: Neurons in one layer of the MLP connect to all neurons in the subsequent layer [25].

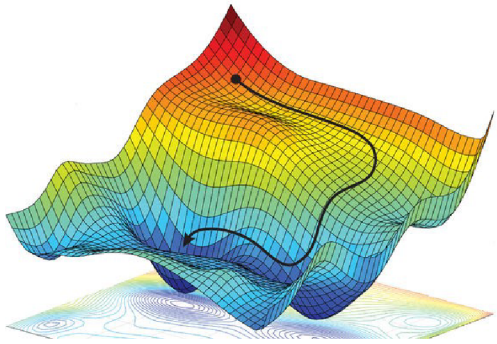


Figure 3.5: During the gradient descent, weights are adjusted in the direction that reduces error [26].

Throughout the training, backpropagation uses a *loss function* that calculates the network's error by comparing its outputs with the correct solutions in the training dataset. The algorithm then looks for the combination of weights which minimizes the loss by using the *gradient descent* procedure. First, it calculates the partial derivatives of the loss with respect to weights and looks at the slope of the tangent. It then adjusts the weights in the direction that reduces the loss, as illustrated in Figure 3.5. The algorithm repeats this process until the error converges to a local minimum [27][28][29][30][31].

The weights alone, however, are not enough to explain how the algorithm arrives at its solution, and a review of the neural network's results by human experts is still necessary to detect and correct possible hidden flaws.

Conclusion

Machine learning enables computers to learn and perform tasks previously only done by humans. Supervised learning is just one category of ML algorithms that tries to find connections between inputs and outputs, and is typically used for classification and regression tasks.

Support Vector Machines classify data by finding a hyperplane that separates the classes in the sample space. Alternatively, an SVR can approximate a function by fitting data within the hyperplane's margins.

Multilayer Perceptron is a type of Artificial Neural Network. It is composed of individual processing units called neurons that are arranged into interconnected layers. The MLP learns by adjusting these connections during training using the backpropagation algorithm.

In the next chapter, I continue with how I collected and prepared the signal strength data that were used to teach these models to locate mobile devices in the GSM network.

4 Dataset

Data play an essential role in the machine learning process. The learning algorithm requires them in order to create the prediction model, and the quality of the dataset can significantly influence the model's performance.

The first part of this chapter introduces some of the utilised tools and software libraries, the second part describes the process of collecting the data and the third part focuses on preparation and analysis of the dataset.

4.1 Used technologies

As a part of this thesis, I have created several Python scripts that experiment with the ML models and assess their suitability for localization based on received signal strength. To perform an analysis and process the collected data, as well as to implement the ML algorithms, I have used several third-party software tools that are described below.

NumPy

NumPy is a Python library that enables efficient handling of multidimensional arrays and matrices and provides a variety of mathematical functions that operate on these arrays. Due to its enhanced performance compared to the standard Python library, it is well suited for scientific computations and other applications that have to manage extensive amounts of data [32].

pandas

pandas is an open-source tool for data analysis and manipulation written in Python. It provides two primary data structures—Series and DataFrames—that are designed for intuitive work with structured tabular data. Among others, pandas can be used to load raw data from CSV or Excel files, handling missing values, data aggregation and database operations such as merging or joining datasets [33].

scikit-learn

scikit-learn is an open-source library built on top of NumPy and pandas that provides simple and efficient tools for predictive data analysis. It implements a range of machine learning algorithms, for instance, Decision Trees, Naive Bayes and Support Vector Machines [34].

Keras

Keras is a high-level API used for fast prototyping with neural networks. It represents the network as a graph of independent modules that can be easily combined to create more complex models. Keras is capable of utilising multiple back-end computation engines such as TensorFlow, CNTK or Theano. It can also take advantage of the GPU to improve performance and decrease training times [35].

OpenCellID

OpenCellID is a community project that collects GPS positions and other information about GSM cell towers all over the world. It is freely accessible and can be used for locating devices or understanding network coverage [36].

4.2 Data collection

The hardware I used for collecting the data consisted of a single *Motorola Moto G6 Play* mobile phone with the Android operating system. The phone was fitted with an O2 SIM card and equipped with a GPS unit. The GPS positioning information was used as ground truth during the experiment.

The signal strength data were obtained using a specialised scanner application *Molotras* developed by Mgr. Jakub Martinka as part of his master's thesis [1]. Once every second, the application requests the GPS position of the device and gathers signal measurements of up to seven surrounding cells.

Molotras records the cells' identification numbers and various properties of the radio signal. However, only the four identifiers—

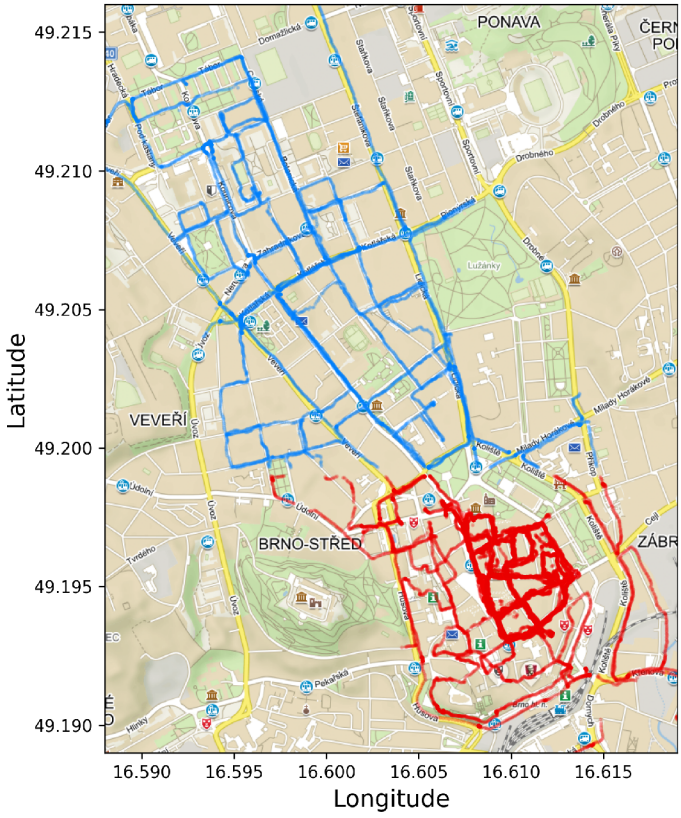


Figure 4.1: I have split the collected measurements into two datasets: city centre (red) and residential (blue). Includes data from [1].

MCC, MNC, LAC and Cell ID—along with the received signal strengths are required to estimate the location of the mobile device.

When the collecting of data is completed, Molotras saves all information into a CSV file. Every measurement consists of a header line with the GPS coordinates, followed by one line for each of the scanned cells.

To collect the measurements, I have ridden a bicycle through the streets of Brno, Czech republic with the scanner application running. Cycling was chosen over walking to provide more extensive coverage of the area.

As signal propagation characteristics vary in different environments, the focus of this thesis was placed on the urban areas of the city centre and the surrounding residential blocks. Areas from open environments that have different propagation properties were removed to improve the performance of the models. Figure 4.1 shows the final area under study after filtering the data. In total, I have collected 3.2 MB of raw data over a period of three days.

Additional data were obtained by including the measurement taken for the Molotras thesis. One of the datasets also covers the city centre of Brno and could be readily joined with the rest of the data. The other dataset was taken in Trenčianska Turná, a village in Slovakia, that served for testing how well the models generalise for other types of environments.

4.3 Data processing

The output of the Molotras application was in the form of a single CSV file with multiple lines per measurement. As the header lines had a different structure from the rest, the raw data had to be parsed with a simple script before loading them into a pandas DataFrame.

The parsing script generates an ID number for each measurement and, along with the GPS location of the sample, attaches it to all records associated with that measurement. All columns are subsequently of the same type, and the tabular data can be worked with further.

The parsed data were then separated into three subsets according to the location in which they were collected. The *centre* dataset covers the city centre and contains 16,240 measurements. The *residential* dataset includes 6,936 measurements from the neighbourhoods north of the city centre, and the *village* dataset comprises 2,387 samples from the Slovakian village.

On average, each measurement contained between five and six individual signal strength recordings. However, due to imperfections of the measuring setup, about 10 percent of the records lacked the cell tower identification and had to be excluded. Table 4.1 provides a more detailed view of the datasets and the number of records per measurement.

Table 4.1: Number of received signal strength records per one measurement.

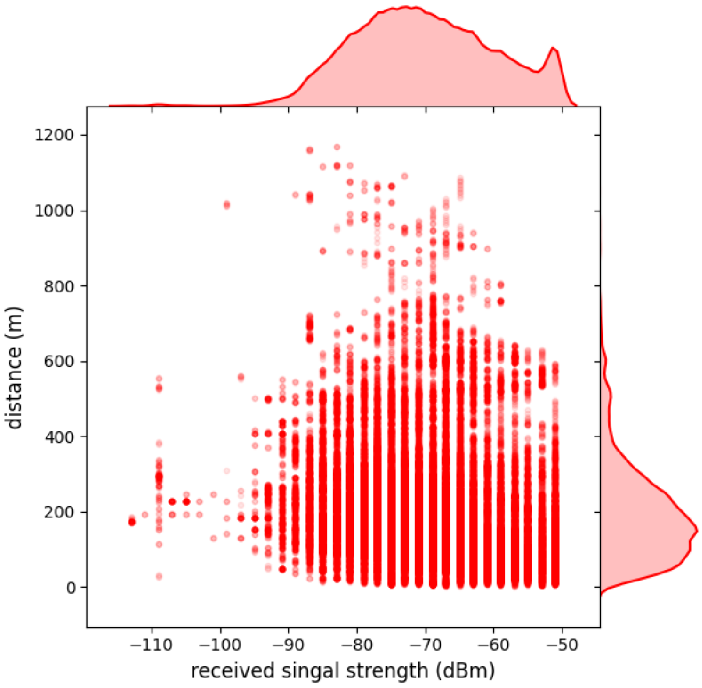
Dataset	Centre		Residential		Village	
Records per measurement	Count	%	Count	%	Count	%
7	5204	32.80	1421	20.49	376	15.75
6	5790	36.49	2445	35.25	786	32.93
5	3259	20.54	1890	27.25	549	23.00
4	1170	7.37	782	11.28	468	19.60
3	349	2.20	279	4.02	205	8.59
2	42	0.27	84	1.21	0	0.00
1	53	0.33	35	0.50	3	0.13
Sum	15867	100	6936	100	2387	100

The next step was to add the BTS towers' GPS coordinates. Fortunately, the OpenCellID project also covers the Czech Republic and Slovakia, and their database could be conveniently incorporated using the `pandas.merge` function that performs a database-style join.

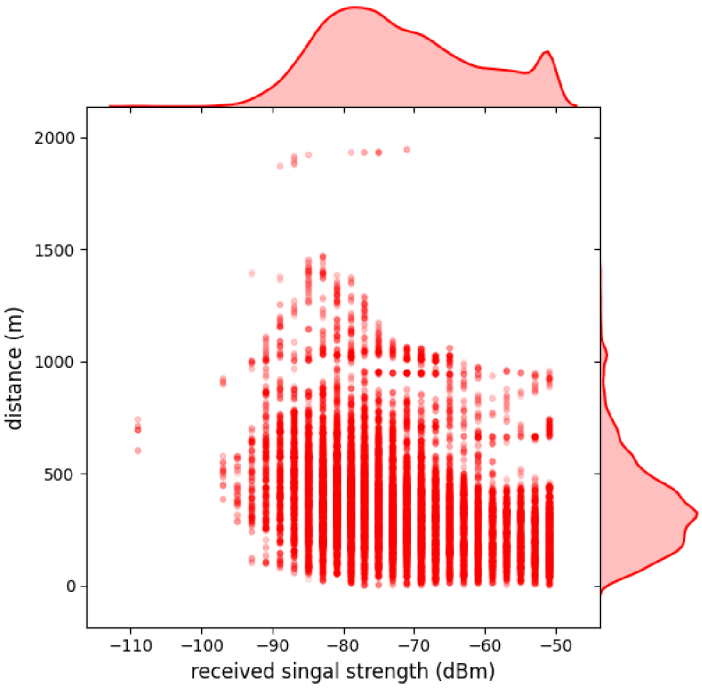
With known positions of the cell towers, distances between the BTSs and the measurement locations could be determined using the haversine formula. The haversine formula calculates the distance of two points on a sphere and can be used to approximate the distance between two locations on Earth.

Figure 4.2 then plots the calculated distance against received signal strength across the three datasets. At first glance, there does not seem to be a clear relationship between the two quantities. Still, the machine learning models may find a more suitable representation of the data and extract useful information.

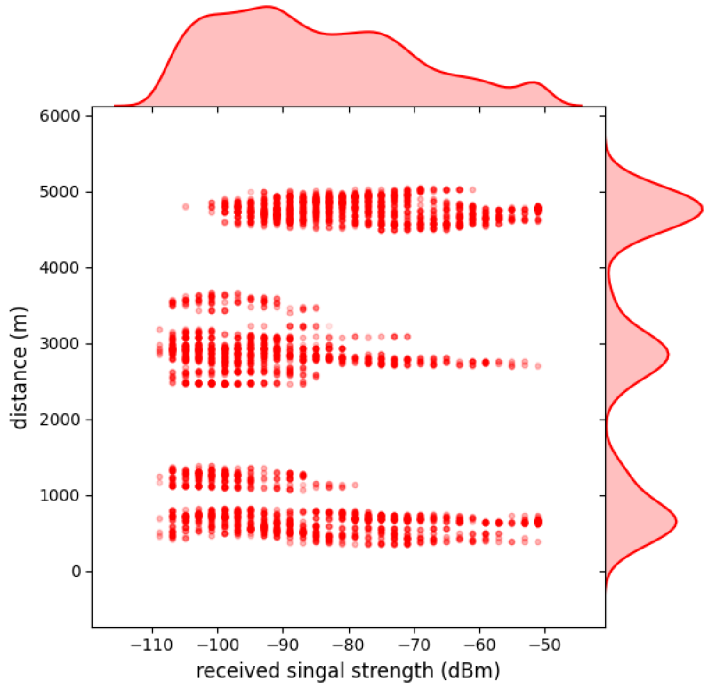
Since the learning algorithms I used in this thesis work with fixed-length inputs, the measurement lengths had to be equalised. It is, however, not apparent how many records each measurement should have. On the one hand, taking fewer records could lead to a loss of information, and on the other, taking more would retain records obtained over a longer distance that are more prone to noise.



(a) City centre



(b) Residential area



(c) Village

Figure 4.2: There does not seem to be a clear relationship between received signal strength and distance.

By experimenting with different values, taking four records per measurement showed a good compromise while retaining over 91 percent of the data since shorter samples were excluded.

The alternative to removing the shorter samples was to generate the missing values. This approach did not provide any significant improvements and on the contrary, could potentially introduce a bias towards the generated values.

For comparison, an additional dataset from the city centre that considers only the one nearest record was prepared, conceivably modelling the propagation of radio signal.

Finally, all datasets were split 60/20/20 into training, validation and testing subsets. The training data served as a reference for teaching the models, the validation set was used to evaluate the models' performance during training, and the testing set was put aside for

the final assessment. The next chapter follows with a more detailed explanation of the training and evaluation.

5 Evaluation

The main objective of the models was to transform received signal strength into distance. The predicted distances were used in multilateration to find the position of the receiver, and the mean localization error was used to evaluate the model.

In the first part of this chapter, I discuss the specifics of training the ML models and tuning hyperparameters of the learning algorithms. In part two, I present the results and examine the models' performance on the prepared datasets. Finally, in the third section, I compare the methods implemented in this thesis with other localization techniques.

5.1 Training

A machine learning model is at its core a mathematical formula. The model learns by adjusting the formula's parameters in a way that fits the learning data. There is, however, another set of parameters affecting the learning process that cannot be derived from the data.

Hyperparameters define higher-level qualities of the model, such as the number of hidden layers of a neural network, that are usually set before the training begins. Choosing the right combination of hyperparameters is done by trying different options and selecting the one that produces the best performing model.

Besides manually searching for the best candidate solution, the most straightforward method of optimizing hyperparameters is Grid Search. Grid Search systematically tests all combinations of hyperparameters and returns the one that yields the best results.

Despite its simplicity, exploring all possible solutions using Grid Search becomes increasingly time-consuming as the number of hyperparameters and their individual choice of values grows. Additionally, the optimal solution may not even lie on the grid but rather in the spaces between the tested values [37][38].

Random Search, in contrast, randomly samples the search space and has been proven to find higher scoring configurations in a shorter time than the Grid Search technique [39]. For the above reasons, I have used Random Search in this thesis to tune the hyperparameters of the SVR and MLP models.

SVR

During the training of the SVR, the considered hyperparameters included the *kernel*, the regularization parameter *C*, the tolerance margin *epsilon*, and the kernel coefficient *gamma*.

The linear and RBF kernels provided the best results among the tested candidates. I excluded the sigmoid and polynomial kernels from the final search, due to poor results of the former and long training times with no significant benefit of the latter.

The C and epsilon parameters were randomly drawn from a uniform distribution in the range 0–32; gamma came from the range 0–4. In general, larger values of C and epsilon, in combination with smaller values of gamma, produced the best results.

MLP

Tuning the hyperparameters of the MLP consisted of adjusting its topology and trying various activation functions.

I experimented with networks of one to three hidden layers with the number of neurons ranging between 4 and 192, decreasing in each consecutive layer. The studied activation functions included the linear, tanh, ReLU and sigmoid activations.

For training the MLP, I used the Huber loss function [40] and the Adam [41] optimizer with default parameters. The effects of other optimizers and learning rates were negligible.

I also examined the option of using Dropout layers and activity regularization to prevent overfitting. Again, with no significant improvements.

Baseline model

I implemented a third, baseline model that estimates the receiver's position equally distant from each BTS. In such a case, the multilateration algorithm computes the weighted centroid of the area given by the cells, assigning each cell the same weight. This model was expected to provide only rough positioning estimates and served for comparison to the ML models.

5.2 Results

All three implemented models were first trained and evaluated in the city centre and residential area. The box chart in Figure 5.1 shows a visual representation of the outcome, and for a more detailed view of the results, you can see the tables in Appendix A.

The median positioning error in the city centre was around 65 meters for all models. However, the accuracy of the SVR dropped significantly when trained on data from the residential parts of the city. The results in the residential area were more balanced, and the median error was approximately 114 meters for all models.

Another approach I studied was modelling the signal propagation function directly. In this case, the models worked with only one signal value at a time instead of using data from all available cells at once. Figure 5.2 illustrates that this method was not as effective, and the median error increased to about 140 meters in the city centre.

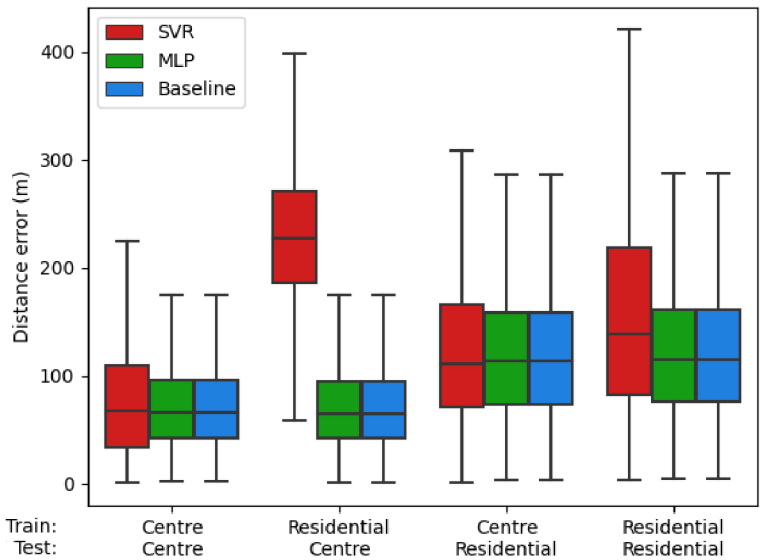


Figure 5.1: The MLP showed identical localization accuracy to the baseline model.

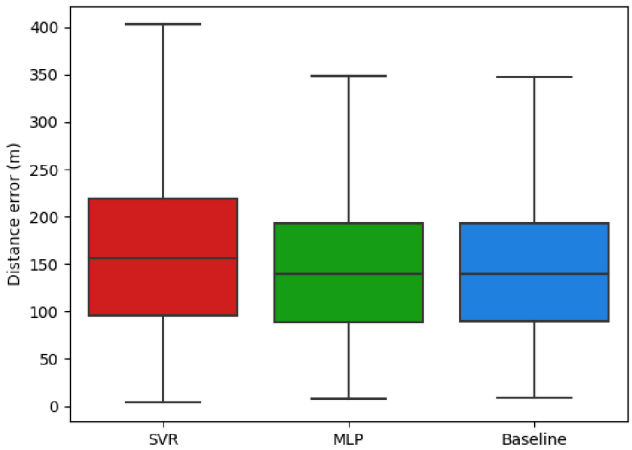


Figure 5.2: All models showed worse performance when working with only one signal value at a time instead of using data from all available cells at once.

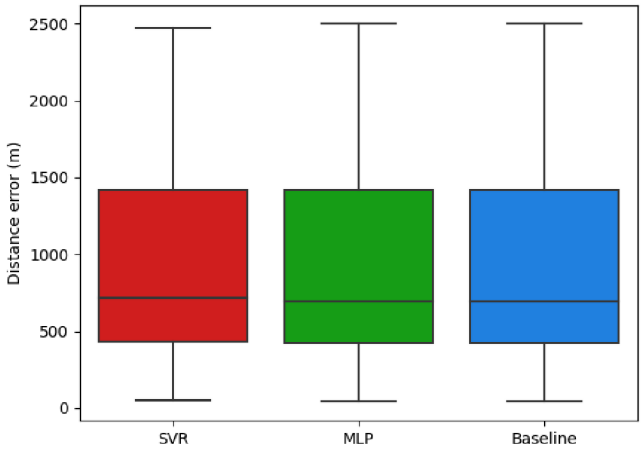


Figure 5.3: The positioning error in the village increased by an order of magnitude.

Finally, to test how well the models generalize to different environments, I evaluated them on the village dataset (see Figure 5.3). As expected, the results differ by order of magnitude. The median positioning error was around 700 meters, with the worst estimates being off by two and a half kilometres.

Discussion

It should be pointed out, that the MLP performed with an identical precision to the baseline model, and the SVR achieved similar results at best. Such an outcome implies that the ML models were not able to filter the noise caused by buildings blocking the signal's path. The models instead learned to mimic the centroid algorithm, as another way of minimizing the localization error. Consequently, the better positioning accuracy in the city centre can be attributed to the higher density of BTS towers, as compared to the residential area or the village.

Upon closer inspection of the MLP, the predictions certainly did not resemble the expected distance values. Moreover, the topology of the network had little significance, as vastly different configuration were able to reach the baseline model's performance.

In conclusion, the ML models I tested in this thesis were not able to take advantage of the multidimensional input and extract useful information that would make them outperform the baseline model. Their performance closely imitated the centroid algorithm, and the localization accuracy could be linked to the density of BTS towers in the tested area.

5.3 Comparison

Tables 5.1 and 5.2 compare the results of the models I have implemented to the localization techniques used in the Molotras thesis—centroid, trilateration and fingerprinting. All methods were evaluated on the same data from the city centre of Brno and the village Trenčianska Turná.

The centroid method computes the arithmetic mean of the cells' coordinates and is comparable to my baseline model. Trilateration uti-

Table 5.1: Fingerprinting in the city centre showed the lowest localization error as compared to other positioning methods that otherwise performed comparably. Data from [1].

Model	Median (m)	Mean (m)	Min (m)	Max (m)
SVR	67.26	78.16	0.42	391.30
MLP	65.27	74.64	1.28	316.87
Baseline	65.28	74.63	1.28	316.62
Centroid	66.34	64.77	17.8	128.37
Trilateration	69.70	69.79	13.75	121.99
Fingerprinting	9.23	16.26	2.22	61.37

Table 5.2: ML models showed higher localization error than the traditional methods when tested on the village dataset. Data from [1].

Model	Median (m)	Mean (m)	Min (m)	Max (m)
SVR	716.24	861.50	52.31	2475.42
MLP	696.23	848.56	40.86	2498.73
Baseline	696.30	848.60	40.93	2498.73
Centroid	605.93	691.69	208.17	2172.08
Trilateration	539.11	587.18	154.97	1889.19
Fingerprinting	49.72	90.43	4.89	306.63

lizes the Okumura-Hata signal propagation model to convert received signal strengths into weights for the multilateration algorithm. Both approaches showed similar results of 65-70 metres median error in the city and slightly outperformed the ML models in the village.

By far, the best performance was achieved with the fingerprinting method. Fingerprinting uses a database of signal strength measurements taken in the area of interest. It compares the currently observed signal readings with the database and locates the receiver based on the most similar reference point. This technique, however, comes with notable disadvantages as creating and maintaining the database is

very time and resource consuming, and has to be done separately for each location.

Other state-of-the-art studies using received signal strength for GSM localization show similar results. For instance, the research [42] achieved 65-metre positioning error in metropolitan areas and showed that an advantage is gained by scanning cells of all available providers. The same holds for [43] where Ibrahim et al. demonstrated that fingerprinting methods reach accuracies around 50 metres even when only using the associated cell tower information.

6 Conclusion

In this thesis, I examined the potential of using machine learning algorithms in GSM positioning.

I have collected a dataset of over twenty thousand signal strength measurements in the city of Brno that was used for evaluating the ML models. The analysis showed a lot of noise in the data caused by buildings blocking the path of the signal—the same readings could be obtained from a nearby cell hidden behind a building, as well as from a more distant cell with a clear line of sight.

The studied models—SVR and MLP—were then expected to learn the relation between received signal strength and distance while taking advantage of signal readings from multiple surrounding cells. For comparison with the ML models, I implemented a third, baseline model that calculated the centroid of the area given by the cells, and provided only rough positioning estimates.

Both SVR and MLP achieved similar positioning accuracy with the median error of 65 metres in the city centre and 115 metres in the residential area. This outcome, however, closely matched the accuracy of the baseline model, which achieved identical results.

Further analysis of the models' performance revealed that they were not able to extract any additional information from the data. Instead, they learned to imitate the centroid algorithm as another way of minimizing the localization error. Consequently, by evaluating the models in different environments, I demonstrated that their accuracy could be linked to the cell tower density in the tested area.

In conclusion, I showed that the tested ML algorithms do not provide an advantage over traditional localization techniques when used to model signal propagation in urban areas.

Possible future research could try to reduce the effects of noise by measuring signal strengths with a clear line of sight to the transmitting radio tower. The fitted ML model could then be used for localization in open environments where buildings do not block the path of the signal. This technique could be directly compared with the Free Space Path Loss model.

Another approach would be to let a neural network learn a map of the area based on signal strength measurements and identities of the

cells. This method would work on the same principle as fingerprinting techniques and would require extensive coverage of the area for training the model. The neural network, however, could prove itself to be a more robust solution that could handle changes in the environment even after the training phase.

Bibliography

1. MARTINKA, Jakub. *Locating mobile phones using signal strength measurements*. 2020. Available also from: https://is.muni.cz/th/asrql/. Master's thesis. Masaryk University.
2. POOLE, Ian. *What is Frequency Modulation, FM* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/radio/modulation/frequency-modulation-fm.php.
3. GOLENIEWSKI, Lillian. *The Electromagnetic Spectrum and Bandwidth* [online] [visited on 2020-05-25]. Available from: https://www.informit.com/articles/article.aspx?p=24687.
4. HENDERSON, Tom. *Reflection, Refraction, and Diffraction* [online] [visited on 2020-05-25]. Available from: https://www.physicsclassroom.com/class/waves/Lesson-3/Reflection,-Refraction,-and-Diffraction.
5. POOLE, Ian. *Free Space Path Loss: details & calculator* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/antennas-propagation/propagation-overview/free-space-path-loss.php.
6. KOSTANIC, Ivica. *Okumura and Hata Macroscopic Propagation Models* [lecture slides]. 2004. Available also from: https://my.fit.edu/~kostanic/RF%5C%20Propagation/Old%5C%20Notes/RF%5C%20Propagation%5C%20-%5C%2007-Okumura%5C%20and%5C%20Hata%5C%20Macroscopic%5C%20Propagation%5C%20Models.pdf.
7. ANANDAMURUGAN, S.; NANDHINI, P. S. *Wireless Networks*. Hauppauge, New York: Nova Science Publishers, Inc., 2016. ISBN 978-1634851787.
8. POOLE, Ian. *Cellular network basics* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/connectivity/cellular-mobile-phone/network-architecture.php.

9. POOLE, Ian. *What is GSM: Primer* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/connectivity/2g-gsm/basics-introduction.php.

10. POOLE, Ian. *GSM Network Architecture* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/connectivity/2g-gsm/network-architecture.php.

11. GISGEOGRAPHY. *Trilateration vs Triangulation – How GPS Receivers Work* [online] [visited on 2020-05-25]. Available from: https://gisgeography.com/trilateration-triangulation-gps.

12. POOLE, Ian. *What is Satellite Navigation SatNav, GNSS* [online] [visited on 2020-05-25]. Available from: https://www.electronics-notes.com/articles/satellites/satellite-navigation-satnav/what-is-satnav-gnss-technology.php.

13. BENSKY, Alan. *Wireless Positioning Technologies and Applications*. 2nd ed. Boston (Mass): Artech House, 2016. ISBN 9781608079513.

14. HEATH, Nick. *What is machine learning? Everything you need to know* [online] [visited on 2020-05-25]. Available from: https://www.zdnet.com/article/what-is-machine-learning-everything-you-need-to-know/.

15. SONI, Devin. *Supervised vs. Unsupervised Learning* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d.

16. DRAKOS, Georgios. *Support Vector Machine vs Logistic Regression* [online] [visited on 2020-05-25]. Available from: https://medium.com/@george.drakos62/support-vector-machine-vs-logistic-regression-94cc2975433f.

17. CHEN, Lujing. *Support Vector Machine — Simply Explained* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496.

18. NG, Andrew. *Support Vector Machines* [lecture notes]. 2018. Available also from: http://cs229.stanford.edu/notes/cs229-notes3.pdf.

19. BHATTACHARYYA, Indresh. *Support Vector Regression Or SVR* [online] [visited on 2020-05-25]. Available from: https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff.

20. SHARP, Tom. *An Introduction to Support Vector Regression (SVR)* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2.

21. KANDAN, Harish. *Understanding the kernel trick.* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/understanding-the-kernel-trick-e0bc6112ef78.

22. SHARMA, Avinash V. *Understanding Activation Functions in Neural Networks* [online] [visited on 2020-05-25]. Available from: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0.

23. MISSINGLINK.AI. *7 Types of Neural Network Activation Functions: How to Choose?* [online] [visited on 2020-05-25]. Available from: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/#section2.

24. SHARMA, Sagar. *Activation Functions in Neural Networks* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

25. WIKIPEDIA. *Artificial neural network* [online] [visited on 2020-05-25]. Available from: https://en.wikipedia.org/wiki/Artificial_neural_network.

26. HUTSON, Matthew. *AI researchers allege that machine learning is alchemy* [online] [visited on 2020-05-25]. Available from: https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy.

27. HANSEN, Casper. *Neural Networks: Feedforward and Backpropagation Explained & Optimization* [online] [visited on 2020-05-25]. Available from: https://mlfromscratch.com/neural-networks-explained/#/.

28. KAPUR, Rohan. *Rohan & Lenny #1: Neural Networks & The Backpropagation Algorithm, Explained* [online] [visited on 2020-05-25]. Available from: https://ayearofai.com/rohan-lenny-1-neural-networks-the-backpropagation-algorithm-explained-abf4609d4f9d.

29. SHANE, Janelle. *Neural networks, explained* [online] [visited on 2020-05-25]. Available from: https://physicsworld.com/a/neural-networks-explained/.

30. ROJAS, Raúl. *Neural Networks: A Systematic Introduction*. Berlin Heidelberg: Springer, 1996. ISBN 978-3540605058.

31. KRIESEL, David. *A Brief Introduction to Neural Networks*. 2007. Available also from: available%20at%20http://www.dkriesel.com/en/science/neural_networks.

32. OLIPHANT, Travis. *NumPy* [online] [visited on 2020-05-25]. Available from: https://numpy.org/.

33. MCKINNEY, Wes. *pandas* [online] [visited on 2020-05-25]. Available from: https://pandas.pydata.org/.

34. COURNAPEAU, David. *scikit-learn* [online] [visited on 2020-05-25]. Available from: https://scikit-learn.org/stable/.

35. CHOLLET, François et al. *Keras* [online] [visited on 2020-05-25]. Available from: https://keras.io/.

36. UNWIRED LABS. *OpenCellID* [online] [visited on 2020-05-25]. Available from: https://opencellid.org/.

37. CAPONETTO, Guilherme. *Random Search vs Grid Search for hyperparameter optimization* [online] [visited on 2020-05-25]. Available from: https://towardsdatascience.com/random-search-vs-grid-search-for-hyperparameter-optimization-345e1422899d.

38. DESCAMPS, Benoit. *Tuning Hyperparameters (part II): Random Search on Spark* [online] [visited on 2020-05-25]. Available from: `https://towardsdatascience.com/hyperparameters-part-ii-random-search-on-spark-77667e68b606`.

39. BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. *Journal of machine learning research*. 2012, vol. 13, no. Feb, pp. 281–305.

40. WIKIPEDIA. *Huber loss* [online] [visited on 2020-05-25]. Available from: `https://en.wikipedia.org/wiki/Huber_loss`.

41. KINGMA, Diederik P; BA, Jimmy. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. 2014.

42. CHEN, Mike Y et al. Practical metropolitan-scale positioning for gsm phones. In: *International Conference on Ubiquitous Computing*. 2006, pp. 225–242.

43. IBRAHIM, Mohamed; YOUSSEF, Moustafa. A hidden markov model for localization using low-end GSM cell phones. In: *2011 IEEE International Conference on Communications (ICC)*. 2011, pp. 1–5.

A Evaluation

Table A.1: When trained on data from the city centre, both SVR and MLP showed similar localization error to the baseline model.

Test set	Model	Median(m)	Mean(m)	Min(m)	Max(m)
Centre	SVR	67.26	78.16	0.42	391.31
	MLP	65.27	74.64	1.28	316.87
	Baseline	65.28	74.63	1.28	316.62
Residential	SVR	110.37	127.22	1.19	677.68
	MLP	113.85	125.53	2.54	624.85
	Baseline	113.75	125.51	2.56	625.21
Village	SVR	716.24	861.50	52.31	2475.42
	MLP	696.23	848.56	40.86	2498.73
	Baseline	696.30	848.60	40.93	2498.73

Table A.2: Training on data from the residential areas significantly impaired the SVR's performance in the city centre.

Test set	Model	Median(m)	Mean(m)	Min(m)	Max(m)
Residential	SVR	138.05	157.89	2.82	684.31
	MLP	114.97	128.09	4.42	569.31
	Baseline	114.79	128.09	4.29	569.33
Centre	SVR	226.85	225.45	2.51	526.22
	MLP	64.57	73.93	0.26	325.74
	Baseline	64.55	73.95	0.15	325.71

Table A.3: The implemented models performed worse when working with only one signal value at a time. Table shows positioning error when evaluated in the city centre.

Model	Median (m)	Mean (m)	Min (m)	Max (m)
SVR	156.75	165.74	3.79	819.89
MLP	139.20	149.81	7.83	815.00
Baseline	139.69	149.67	9.45	816.79

B Archive structure

The electronic archive *thesis.zip* submitted with this thesis contains the following files:

- **cells/230.csv** and **cells/231.csv**- OpenCellID database files covering the Czech Republic and Slovakia respectivelly [36]
- **data/cell_data.csv** and **data/sk_cell_data.csv** - signal strength measurements (Brno + Trenčianska Turná) [1]
- **data/my_data.csv** - signal strength measurements I have collected (Brno)
- **main.py** - main script that runs the experiment, run via ./main.py
- **evaluation.py**, **models.py**, **parse.py**, **preprocessing.py** and **trilateration.py** - Python scripts that execute individual parts of the experiment
- **out.txt** - example output of `main.py`
- **requirements.in** and **requirements.txt** - Python requirements generated by `pip-compile`