Bachelor Thesis

RSSI-based Indoor Localization Using Machine Learning and Deep Learning

---

by

Máté Nagy

(2697010)

First supervisor: Kees Verstoep
Daily supervisor: Kees Verstoep
Second reader: ?

June 12, 2023

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science

RSSI-based Indoor Localization Using Machine Learning and Deep Learning

Máté Nagy
Vrije Universiteit Amsterdam
Amsterdam, The Netherlands
nagymatepeter@student.vu.nl

ABSTRACT

Context. Write this at the end
Goal. Write this at the end
Method. Write this at the end
Results. Write this at the end
Conclusions. Write this at the end

1 INTRODUCTION

In the recent years navigation has gone through some great advancements due to the rise of technology. Global Positioning Systems (GPS) have advanced greatly and have been a crucial part of our daily lives helping us navigating around our world. However, when it comes to navigation in indoor environments GPS-based navigation systems are not as reliable as in outdoor environments due to the buildings' blocking and weakening effect that is caused by their construction materials. This is provides a great disadvantage to individuals attempting to get around in shopping malls, hospitals, airports, large office buildings or university buildings.

To overcome these limitation researchers and companies have been exploring what are the possibilities to provide accurate indoor localization in indoor environments. Although there does not exist one single standard yet many possible solutions are available. Some of these include the use of technologies such as Bluetooth Low Energy (BLE), Ultra-Wideband (UWB), Radio Frequency Identification (RFID) and Wi-Fi that are widely available in most of the larger building. Some of the techniques that can utilize these technologies are RSSI (Recieved Signal Strength Indicator), Time of Flight (ToF) and Trilateration. RSSI measures the power of radio signal emitted by the beacon or wireless device, while Time of Flight measures the time it takes for a signal to travel from the transmitter device to the receiver device. Trilateration on the other hand works by determining the distance measured between 3 fixed points to provide an accurate location of the user. Even though these methods are able to give a location estimate they are not reliable and accurate enough for indoor environments. Therefore, applying technologies such as machine learning and deep learning that are a subset of artificial intelligence could yield in better accuracy of predicting a device's position. Machine learning and deep learning have been gaining a substantial amount of attention and development over the past decade.

This research paper aims to investigate the impact of applying machine learning and deep learning to accurately determine the indoor location of a mobile device. We start by building a basic indoor RSSI-based localization model using Wi-Fi and compare its precision with the literature. Then we will investigate the impact of the following modifications of our basic model:

- the use of machine learning versus deep learning models;
- the impact of the orientation of the mobile device;
- the use of absolute AP location information versus a fingerprinting approach;

By addressing these questions this study aims to provide an answer to the accuracy of using machine learning and deep learning to solve the challanges of indoor localization.

The thesis start by reviewing related literature, then describing the methods used to collect the data, how the models were created and eventually how the output location was created and then compared to other models. After that the results will be presented along with an experiment followed by the discussion and conclusion. Lastly the possible ways of future work will be described.

2 LITERATURE REVIEW

In this section an overview will be provided on related studies and methodologies that have contributed to the domain of indoor localization using the WiFi RSSI method. By examining the relevant literature, we have the intention to highlight the key findings of other studies.

The study explored the potential of using a fingerprinting method on WiFi RSSI values from various access points where deep learning was utilized. Their model demonstrated an accuracy of 95.95%. The findings of the study seem promising and suggest that using deep learning for predicting indoor positions can be a good approach.[3]

This project investigates the impact of using WiFi-based indoor positioning with the main focus being on the device's orientation. The study uses a fingerprinting approach on RSSI gathered from WiFi Access Points to determine the device's location and orientation within 1.8 meters in an elderly center. The results show that the orientation and the location of the device can be distinguish with their approach.[1]

Another study explored the impact of using a trilateration method and the use of a deep learning model where both approaches used a dataset containing RSSI values amongst other features. The research concludes that the application of deep learning had a significant impact on the accuracy of prediction the device's position in the university building over the naive trilateration approach. It was highlighted, however, that the deep learning model is less generalizable and computationally extensive. [2]

3 METHOD

This section provides a description of how the data was collected and preprocessed. Furthermore, it will describe the fine tuning and training of the machine learning models created as well as the hyperparameter tuning and training of the deep learning model.

3.1 Data Collection Application

For the purpose of data collection an Android application was created using Android Studio. To test the application an Android emulator with Android 10 (API level 29) was used and a Google Pixel 2 that also has Android 10 as its operating system. Even though there are newer versions of Android available, this device only supports Android up to version 10. Furthermore, it was important to use Android 10 for this application since this version introduced a new developer option in the phone's settings to turn off throttling for local testing of the WiFi scanning capabilities provided by the WifiManager API. The application was able to scan for WiFi signals from surrounding devices and gather information about them, most importantly the RSSI of each device. This data later was written to the test phone's storage in csv format for later use.

3.2 Data Collection

The data collection process involved the use of the Android application made for the data collection. The data collection process took place at the Vrije Universiteit Amsterdam on the 5th and 6th floors of the NU building. These two floors were selected due to their versatility of room sizes and shapes, isles and the open study areas. There are 29 access points on the 5th floor and 34 acccess points on the 6th floor of the building. On the 5th floor 30 measurement points and on the 6th floor 31 measurement points were selected. These points were distributed through the floors to have data on most of the rooms and spaces. At each measurement point 60 WiFi RSSI scans were made resulting in a total of 3660 instances. Each instance that represent a scan involved information on the RSSI of a given access point. Access points that are too far for the WiFi scanner to pick up their signal were represented with a 1 in the dataset and the ones that were close enough to the scanner had their RSSI in the dataset. Moreover, each instance held the location information of the measurement point represented by X, Y and Z coordinate values of the building. The X and Y coordinate values represent 1 meter distance in the building and the Z value represent the floor where the measurement was taken. Additionally, to record the directionality of the device after every 15th scan the device was oriented towards the next cardinal direction starting from north. The intention of collecting this data was to increase the effect of machine learning and deep learning on the data therefore increasing the accuracy of the localization.

3.3 Data preparation and preprocessing

When the data was collected it was transformed using the programming language Python and with the help of several Python libraries such as NumPy and Pandas. Since the data was in many separate csv files they had to be organized into one dataframe containing 3660 instances and 327 columns. Each row contained information about each AP's RSSI in the whole building and the location of the measurement. Then the two floor's coordinate values could be added to the dataframe to enrich the dataset with additional features. Furthermore, the directionality of the device had to be represented by numerical values therefore when the device was facing north the directionality was expressed as a 0, when facing east with the value 90, when facing south with the value 180 and lastly when facing towards west with 270. After all the data was prepared, a scaling procedure was implemented to normalize the values within the range of 0 and 1. Scaling is the process of transforming the values of the features of a dataset to a standardized range such that they are easily comparable. It allows for the prevention of skewed results and for a better performance of machine and deep learning models. Finally the dataset got split into a training set and a test set with sizes of 80% and 20% respectively.

3.4 Machine learning Hyperparameter Tuning and Training

After the data has been prepared, scaled and split, 4 different machine learning models were created. These models were created with the use of scikit-learn, an open source python library used for data analysis and machine learning solutions. Since the basis models' goal is to predict numerical coordinates for an indoor environment a regression-based approach was used on all 4 methods. The 4 methods include decision tree algorithm, random forest algorithm, support vector machine algorithm and K-nearest neighbor algorithm. The decision tree algorithm is a supervised machine learning method that uses a tree-like structure to make predictions recursively based on its input. Each node in the tree represents a decision and the leafs of the node describe its predictions. The random forest algorithm combines the output of multiple decision trees to make its predictions. It randomly selects a number of trees to create a forest of trees which ultimately improves the performance and accuracy of the model. The support vector machine algorithm (SVM) works by attempting to find a hyperplane in an N-dimensional space that can separate the data into two distinct classes such that the classification error is minimized. The K-Nearest Neighbors algorithm (KNN) operates on the principle of assigning a predicted value to the data point based on its k nearest neighbors in the training set. This number is crucial for the performance of the model. In addition to the mentioned models a multi output regressor was used to allow the models to have 3 outputs.

After the models were created the hyperparameter tuning could begin. For this purpose grid search was used to optimize the performance of our models. When the best hyperparameters got determined the models got adjusted with them to achieve a better performance and accuracy on the data.

3.5 Deep learning Hyperparameter Tuning and Training

Subsequent to fine tuning and training the machine learning models a deep learning model was created with the use of Keras, an open source library to build artificial neural networks that acts as an interface for the TensorFlow library. For determining the best hyperparameters Keras Tuner was used that is a hyperparameter tuning framework. This process involved defining the hyperparameters that were the number of layers, the number of neurons, if the network should have batch normalization or not, the dropout rate, the learning rate, the number of epochs as well as the batch size. The activation function of the search space was set to reLU (Rectified Linear Unit) that sets all of the negative values it gets to zero and leaves the positive ones unchanged. For the output layer linear regression was used as its activation function. When the search space was defined with the range of values of the hyperparameters a tuner had to be chosen. For this purpose Bayesian optimization was used that operates on a statistical principle and utilizes an exploration-exploitation method to find the most optimal hyperparameters. The main objective of the optimizer was to minimize the Mean Squared Error (MSE) that measures how close the regression line gets to the data points of the dataset. Next the tuner was configured to have a maximum of 5 trials to test a set of hyperparameters with 3 executions per trial.

After the search for the hyperparameters the model was trained on various epochs and batch sizes with the option of shuffling the data at each epoch. Once this training was done the model was retrained with the best hyperparameters that were achieved from the previous training.

Lastly the model got evaluated on its performance and using the test set predictions were made for further evaluation to calculate its accuracy with the R-squared statistical measure.

In the end the predictions with the actual results got visualized in 3 dimensional space and the training and validation loss was plotted against each other.

4 RESULTS

In this section, we present the performance evaluation of our approaches. The goal is to assess the accuracy and effectiveness of our models for accurately predicting the indoor position of the device from the RSSI values retrieved from the access points installed in the building. The results will be evaluated on the R-Squared (R2) or R2 score of the model which is a statistical measure describing how well a regression based model is capable predicting the outcomes. The other metric that will be used is the Mean Squared Error (MSE) that is a measure used to determine how close the regression line is to the data points.

4.1 Basis Models

First, we will evaluate the basis models performance of the four machine learning models namely random forest, decision tree, support vector machine, K-nearest neighbors and the deep learning model. To compare the models we report the mean and standard deviation of the test MSE and R2 score over five runs of the models in Table 1.

Model	MSE	R2
Random Forest	0.00104 ± 0.00001	97.342% ± 0.027
Decision Tree	0.00271 ± 0.000096	92.276% ± 0.265
Support Vector Machine	0.00401 ± 0	61.8262% ± 0
K-Nearest Neighbors	0.00204 ± 0	94.85853% ± 0
Deep Learning	0.002585 ± 0.00026	92.4966% ± 1.2301

Table 1: Mean and standard deviation of MSE and R2 score of the basis models.

As it can be seen from Table 1, Random Forest outperforms the other models with the highest R2 score of 97.342% and the lowest MSE being 0.00104. This demonstrates that Random Forest is the most suitable model for a a purely RSSI-based solution that does not include addition features. The second best model is K-Nearest Neighbors with its R2 score of 94.85853% and its MSE of 0.00204. This suggests that K-Nearest Neighbors is also capable of accurately providing indoor location estimates. The deep learning model seems to be able to provide a somewhat accurate estimate, however its standard deviation on the R2 score is quite high compared to the other models. With a 92.276% R2 score and a 0.00271 MSE the Decision Tree has quite similar results compared to the Deep Learning model except for their standard deviations which are by far different. The Support Vector Machine has the poorest performance with a R2 score of 61.8262% and a MSE of 0.00401 which is quite low compared to the other models. This result is mostly due to the fact that the SVM was not able to distinguish between different floors shown on figure 1 taken from the first run.

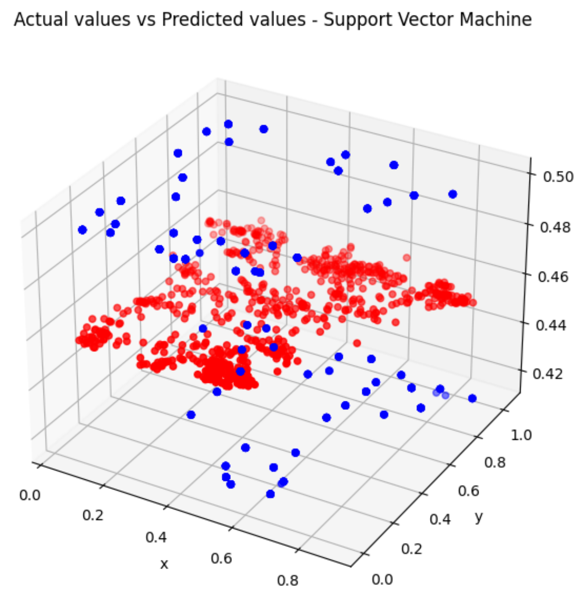

Figure 1: Actual values versus the predicted values of SVM plotted in 3D space.

As can be observed from the presented evidence, all the values that SVM predicted are on the same Z value which is in-between the two floors. This is an immense disadvantage of using SVM for indoor localization.

4.2 Models with Orientation

To evaluate the performance of the models when the orientation of the device was also included as a feature in the dataset, we report the results of the models in Table 2.

As indicated by Table 2, Random Forest has the best performance with the highest R2 score of 97.1626% and the lowest MSE of 0.00114. The second best performing model for this approach is the Support Vector Machine with a R2 score of 95.2857% and a MSE of 0.0017. K-Nearest Neighbors still performs quite well, however not as good as the aforementioned models. It has a R2 score of 94.6398% and a MSE of 0.00204. The deep learning model has slightly improved compared to the basis model, however, the machine learning models

Model	MSE	R2
Random Forest	0.00114 ± 0	97.1626% ± 0
Decision Tree	0.00644 ± 0.0018	84.9514% ± 2.8608
Support Vector Machine	0.0017 ± 0	95.2857% ± 0
K-Nearest Neighbors	0.00204 ± 0	94.6398% ± 0
Deep Learning	0.00196 ± 0.00005	93.82% ± 0.2611

Table 2: Mean and standard deviation of MSE and R2 score of the models including the orientation of the device.

seem to outperform it in this approach as well. The worst performing model in this case was the Decision Tree with a R2 score of 84.9514% and a high standard deviation of 2.8608 and a MSE of 0.00644.

4.3 Basis Model versus Model with Orientation

This section presents the key findings of the result obtained from the basis models versus the results from the approach including device orientation. The approach including device orientation showed promising performance results across its models with Random Forest achieving a R2 score of 97.1628% and a 0.00114 MSE closely followed by SVM with a 95.2857% R2 score and a 0.0017 MSE. The SVM in the basis models had a 61.8262% R2 score which is significantly lower than what was achieved by introducing device orientation. K-Nearest Neighbors had a slight decrease in performance compared to the one from the basis models, however, its not significant. The Decision Tree of the orientation based approach in contrast with the basis Decision Tree model had an almost 8% decrease in performance on the R2 score and the MSE increased from 0.00271 to 0.00644. Lastly the deep learning model of the basis model had a 92.4966% R2 score which in contrast to the device orientation approach was more than 1% less accurate.

4.4 Models with Access Points

In this section the models with the coordinate values of the access points on the 5th and 6th floor of the building included are evaluated. To report the results the mean and standard deviation of the test MSE and R2 scores are compared over 5 runs in Table 3.

Model	MSE	R2
Random Forest	0.00109 ± 0.000014	97.2644% ± 0.0588
Decision Tree	0.00339 ± 0.0005	91.5676% ± 0.6321
Support Vector Machine	0.00207 ± 0	94.2304% ± 0
K-Nearest Neighbors	0.00204 ± 0	94.8585% ± 0
Deep Learning	0.00202 ± 0.00009	93.9234% ± 0.16198

Table 3: Mean and standard deviation of MSE and R2 score of the models including the Access Points of 2 floors.

Looking at Table 3, it is noticeable that Random Forest has the best performance once more with a R2 score of 97.2644% and a MSE of 0.00109. The second best model of the approach including access points is the K-Nearest Neighbors with a 94.8585% R2 score and a 0.00204 MSE. The Support Vector Machine compared to KNN has almost the same performance differing only by a small amount on both metrics. The deep learning model ranks on the fourth place in this scenario as well with a 93.9234% R2 score and a 0.00202 MSE.

4.5 Basis Model versus Model with Access Points

This section presents the results of the basis model compared to the model that also includes the location of the access points on the 5th and 6th floor of the building. The Random Forest is the highest performing model in both approaches with the access point locations included being 97.2644%, which is a slight decrease compared to the basis model, however, it is insignificant. The second best model of both approaches is the K-Nearest Neighbors that has a 94.8585% R2 score and a 0.00204 MSE. In fact, both approaches seem to have the same result regardless of the addition of the access points. The SVM of Table 3 shows a 94.2304% R2 score that is significantly greater than the results of the SVM in Table 1 being 61.8262%. The results of the deep learning model presented in Table 3 show a 93.9234% R2 score and a 0.00202 MSE while the deep learning model of Table 1 is slightly worse with a 92.4966% R2 score and a 0.002585 that. Lastly, the Decision Tree of Table 1 shows a 92.276% R2 score with a 0.00271 MSE that performed slightly better than of Table 3 with R2 score 91.5676% and MSE 0.00339.

5 DISCUSSION

In this study, 4 machine learning models and a deep learning model with 3 different approaches got evaluated and compared to accurately determine the indoor localization of the device. To determine which approach and which model performs the best the models were evaluated and valuable insights were gained about their performance.

Show the findings, compare ML vs DL, orientation and AP

5.1 Limitations

Time
Computational Resources
Wifi Scanner API limitation

6 CONCLUSION

In the course of this study we collected vast amount of RSSI data and other additional information and evaluated the performance of each machine learning and deep learning model in various approaches. The results showed us that using machine learning or deep learning has a great potential and can be extremely advantageous to accurately predict the position of devices in indoor environments based on Wi-Fi RSSI data.

7 FUTURE WORK

using multiple devices
directionality with different devices
trajectory based approach
using environment information

REFERENCES

[1] Mohd Nizam Husen and Sukhan Lee. 2014. Indoor human localization with orientation using WiFi fingerprinting. (2014). https://doi.org/10.1145/2557977.2557980

[2] Rik Timmer. 2021. RSSI-Based Indoor WiFi Localization Using Deep Learning. (2021).

[3] Gülsüm Zeynep Gürkaş Aydın Ahmet Sertbaş Zeynep Turgut, Serpil Üstebay. 2018. Deep Learning in Indoor Localization Using WiFi. (2018).