LSTM-based Indoor Localization with Transfer Learning

Martijn Brattinga
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
m.brattinga@student.utwente.nl

ABSTRACT

Localization techniques are the basis for applications such as pedestrian navigation, warehouse asset tracking, and augmented reality. Indoor localization techniques based on the Received Signal Strength Indicator (RSSI) exist that take advantage of existing infrastructure, such as WiFi routers and smartphones, present in practically every building in our modern society. To overcome the challenges caused by the attenuation and scattering of wireless signals in indoor environments, machine learning approaches to improve fingerprinting localization have been studied. Recurrent Neural Networks (RNNs), and in particular Long Short-Term Memory (LSTM), have been found to be effective for indoor localization. Deploying fingerprinting localization with machine learning, however, is expensive. As every environment has different characteristics, a vast amount of data has to be collected for every new environment to train the model on, in order to obtain adequate accuracy. Transfer Learning (TL) techniques have been developed to reduce the amount of required training data for RNNs, lowering deployment costs, however this has not been a topic of research in LSTM-based indoor localization yet. This paper proposes an LSTM-based fingerprinting localization architecture, that utilizes Transfer Learning techniques to provide high accuracy and little deployment costs. This makes indoor localization cheaper and easier to use, enabling it to become more broadly available. A prototype of the proposed model has been made to evaluate the accuracy and deployment costs. The proposed TL techniques significantly improve LSTM-based fingerprinting and reduce deployment costs for indoor localization.

Keywords

Long Short-Term Memory, Fingerprinting, Transfer Learning, Indoor localization, Recurrent Neural Network

1. INTRODUCTION

The demand for accurate indoor localization has become higher over the past decades. The user's location is the basis for applications such as pedestrian navigation, asset tracking, and augmented reality. In outdoor environments, the Global Positioning System (GPS) can provide the user with this location. In indoor environments, however, GPS does not always suffice.

Requirements for indoor localization differ from outdoor localization. Indoor environments are smaller than outdoor environments, and in general, objects are closer together. Less accuracy is needed to locate a building in a street than to locate a door in an office corridor. The GPS cannot provide such a high accuracy indoors, as the wireless signals used are attenuated and scattered by construction walls and roofs, which heavily influences the localization precision. Therefore, another localization method based on WiFi, and in particular on the Received Signals Strength Indicator (RSSI) [14], has gained increasing interest as an alternative in indoor environments.

RSSI fingerprinting localization uses existing infrastructure such as WiFi routers and smartphones with WiFi capabilities. Every location in an environment has a unique combination of distances to neighbor routers, and signal strength depends on the distance between sender and receiver. This implies that at each location, a unique set of received signals, the fingerprint, can be observed. The localization consists of two phases: the offline and the online phase. In the offline phase, fingerprints are gathered for a large number of locations and stored in a database, called the radio map. In the offline phase, a fingerprint is observed at an unknown location. This fingerprint is compared to the radio map, which results in a predicted location.

Wireless signals suffer from attenuation and scattering, making the RSSI vary over time. This makes the process of matching the fingerprint to the radio map not straightforward anymore. Several machine learning techniques have been used in combination with WiFi fingerprinting to overcome this challenge. During the offline phase, these algorithms build an understanding of the environment taking into account attenuation and scattering. This machine learning is done by analyzing a lot of data from the environment. The better the model is a representation of the physical environment, the better the prediction of the location will be.

The machine learning algorithm used in this paper is called Long Short-Term Memory (LSTM), a Recurrent Neural Network (RNN). Traditional RNNs work very well on sequence problems, but they might suffer from vanishing gradient, and exploding gradient problems, which makes them hard to train properly [7]. LSTMs try to solve this problem. Since RSSI sequences are temporally correlative [12], LSTM is a promising method for RSSI fingerprinting for indoor localization.

Every building has different characteristics in terms of wireless signal propagation. The model has to be trained on each environment it is used in, as it has to represent the characteristics of that particular environment. Data collection and processing is an expensive process, and the

35th Twente Student Conference on IT July 2nd, 2021, Enschede, The Netherlands.

requirement to carry it out for every new environment makes this a disadvantage of machine learning-based fingerprinting.

In this research, Transfer Learning techniques to teach a Recurrent Neural Network (RNN) about one environment by using the knowledge of another environment will be studied. Transfer learning is a technique that aims to improve the learning of the target predictive function in the target domain, using knowledge from the source domain [6]. This will lower the amount of data required for training in new environments, and will thus decrease deployment costs.

While quite some research has been done into the use of Long Short-Term Memory as a promising approach for indoor localization [12], as well as Transfer Learning for reducing resources required for the learning phase [6], these two techniques have not been examined together.
In this paper, an LSTM-based fingerprinting approach using Transfer Learning is proposed that reduces deployment efforts for accurate indoor localization.

1.1 Research question

The problem statement can be specified with the following research question:

- **RQ1:** How can Transfer Learning decrease the deployment costs of LSTM-based indoor localization while maintaining accuracy?

To answer the research question, the supplementary questions below will be addressed:

- **RQ1.1:** What knowledge of a trained LSTM-based model of an environment can be used for training on another environment using Transfer Learning?
- **RQ1.2:** How accurate is LSTM-based indoor localization with limited training data using Transfer Learning techniques?

These sub-questions will be answered by literature research, implementing a prototype, and evaluating the prototype's performance. The accuracy for indoor localization will be defined by the mean absolute error, the distance between the actual and the predicted location.
This research is expected to contribute with an LSTM-based fingerprinting localization effort with Transfer Learning techniques that have reduced deployment costs and similar performance as the state of the art.
The rest of this paper is organized as follows. In section 2 related work on fingerprinting for indoor localization, LSTM-based indoor localization, and Transfer Learning is reviewed. After this, the proposed architecture with Transfer Learning is explained in 3. An experiment that evaluates the performance is conducted and analyzed, which is shown in section 4. Finally, in section 5 this paper is concluded.

2. RELATED WORK

In this section, related work on fingerprinting for indoor localization, LSTM-based indoor localization, and Transfer Learning will be reviewed.

2.1 Fingerprinting

Indoor localization using WiFi was a topic on the IEEE Data Mining Contest back in 2007 [13], which brought up several approaches for predicting locations based on WiFi, also taking into account variability of signal characteristics over time. Multiple approaches have been taken since then, with WiFi RSSI fingerprinting being the most popular. Several aspects of RSSI fingerprinting have been explored in [14]. This work explains the general idea of the offline and online phases well. Yiu et al. also describe the influence of architectural parameters such as the density of access points and the density of the radio map.
Several algorithms for the online phase have been explored, where Youssef et al. proposed a solution based on probability distributions [15]. Later on, K nearest-neighbours [11] became the most popular algorithm to determine a location based on RSSI fingerprints. These algorithms certainly proved that fingerprinting localization was promising, but scattering and attenuation still were a use challenge.

2.2 LSTM

When machine learning became more popular, the use of Deep Learning for fingerprinting localization was a new topic of research [4]. The idea was to lower the workforce of deploying an indoor localization infrastructure, as less manual work was needed with deep learning in comparison to previous methods. One particular type of Deep Learning used for indoor localization is based on Convolutional Neural Networks (CNN) [9]. Song et al. managed to create a model with a high success rate on public data sets. However, CNNs do not use the full potential of sequence data.
The Long Short-Term Memory (LSTM) architecture, on the other hand, is very capable of handling sequence data. LSTM has been around for more than two decades and recently became state of the art in many fields [2]. Greff et al. discuss the internals of several LSTM architectures, as well as several parameters and applications. Sahar et al. found LSTM to be an efficient approach to fingerprinting localization[8]. They observed that bi-directional LSTM outperforms other machine learning approaches by a considerable margin. In [1] the focus is mainly on local feature extraction to use in the LSTM fingerprinting approach, which also outperforms other techniques. Xu et al. explored the same concept of LSTM-based RSSI fingerprinting, but this time with Bluetooth [12]. One should note that Xu et al. used simulations to evaluate the performance, thus real-life performance might differ.
The research mentioned above proves that LSTM-based fingerprinting is a promising approach to indoor localization. The main reason is that RSSI sequences are temporally correlative, and LSTM is efficient for processing sequential data [12]. LSTM consist of memory cells, which maintain their state over time, to use long-term dependencies [2]. An LSTM cell has an input, forget and output gate with separate activation functions, to manage state flow. The design of the LSTM architecture makes the LSTM solve the vanishing and exploding gradient problems [7].
In previous research, various hyperparameters are evaluated for RSSI fingerprinting. Sahar et al. found that a stacked LSTM with two layers, each with 50 cells, has a high accuracy [8]. Furthermore, Sahar et al. also explained that the input of the LSTM should be normalized to increase the effectiveness of the training. These values seem reasonable, and this research will use them as a starting point for the model used in this research.

2.3 Transfer learning

The concept of Transfer Learning in its various forms has been a topic of research for more than a decade[6]. Pan et

al. discuss the various types of Transfer Learning in their survey, as well as applications of the technique. They define Transfer Learning as the technique that aims to help improve the learning of the target predictive function in the target domain, using knowledge in the source domain, where either the source and target domains are different, or the source and target tasks are different. The goal of Transfer Learning is to reduce the amount of data required for training a machine learning model in new domains or on new tasks.
More recent research also focuses this Transfer Learning knowledge on indoor localization [10]. Sorour et al. proposed a scheme for joint indoor localization and radio map construction that can be deployed with a limited calibration load. Zhang et al. suggest a Fussy Clustering-based approach with a Manifold Alignment Transfer Learning technique [16], that shows decent accuracy. The downside of this approach is the big time complexity. The problem of an environment changing over time, for instance, because of temperature changes or variance in crowdedness, is a topic of research in [17]. Zheng et al. make it possible to transfer knowledge from a model to reduce calibration effort for other points in time, in the same environment. This research shows that Transfer Learning techniques can be applied for indoor localization, but it does not address the large amount of deployment effort required to localize in a new environment. The variance of environmental characteristics of wireless signals per environment is the main topic in [5], where Pan et al. propose an approach to transfer data from a trained model on one area to another area. Pan et al. solve two problems for Transfer Learning in their work: what to transfer and how to transfer. Previous work on Transfer Learning for indoor localization shows that the technique is promising, and leaves room for improvement by combining it with other state-of-the-art Deep Learning techniques.

As shown, research on several aspects of (LSTM-based) indoor localization and Transfer Learning has been conducted, but these concepts have not been combined yet. The literature can be used to understand the various aspects, which will be required to combine everything.

3. APPROACHES

This section will explain the localization and Transfer Learning process. We take $e \in A, B$ to represent the environment, where A is the source environment, and B is the target environment.

3.1 Fingerprint localization

Fingerprinting localization consists of two phases, the online and the offline phase. These phases are shown in figure 1. In the offline phase, WiFi and Bluetooth signals from sending nodes are measured at several known locations. These fingerprints are, labeled with their locations, put in a database. This database is called the radio map. In the online phase the RSSI values of all nodes in that environment are observed, at an unknown location. This fingerprint is compared to the radio map, from which the location corresponding to this fingerprint can be retrieved. In this research, the database is not a traditional lookup table, but a machine learning regression model, as described in 3.3. This model outputs the x and y coordinates based on the given input, which should correspond with the given fingerprint.

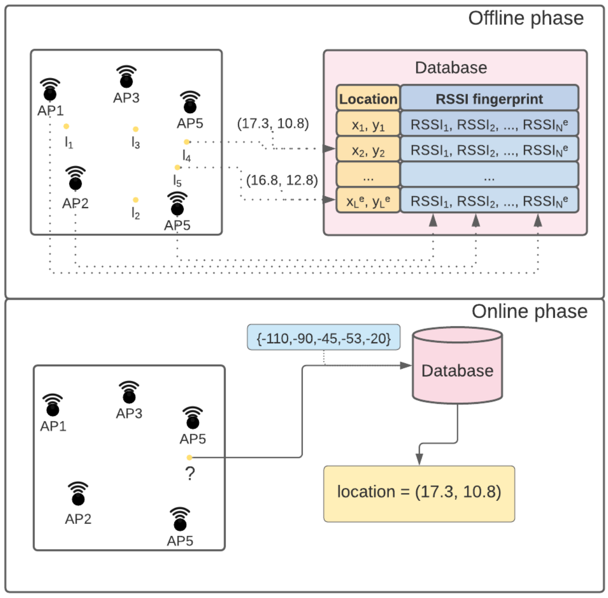


Figure 1: The offline and online phases of the fingerprinting process

3.2 Problem Formulation

An environment consist of N^e sending nodes, being Access Points (APs) or Bluetooth beacons. A sending node can be individually indicated as n_i^e, with $i \in \{1, 2, ..., N^e\}$. It should be noted that N^A does not have to be equal to N^B. In an environment measurements are taken in $L^e = (x, y)$ different locations, individually indicated as l_i^e, with $i \in \{1, 2, ..., L^e\}$. There are $M_{l_i^e}$ different measurements taken for each location, after each other as a sequence. For simplicity, in this research $M_{l_i^A}$ is the 30 for every $i \in \{1, 2, ..., L^A\}$, and $M_{l_i^B}$ is 15 for all $i \in \{1, 2, ...L^B\}$. One measurement contains both the x and y coordinates, as well as the received signal strength of all sending nodes $(-110.0 \leq RSSI_{n_i^e} \leq 0)$ in the environment. If no signal is received from a sending node, the value is set to -110, the minimal value. A measurement for location l_i^e is indicated with $S_{l_i^e j} = \{x, y, RSSI_0, RSSI_1, ..., RSSI_{N^e}\}$.
E^x represents the accuracy of our machine learning model, which is the mean absolute error between predictions and actual locations in meters. This research uses Transfer Learning as described in section 3.4 to provide a model where L^B is significantly smaller than L^A, while E^B is not significantly bigger than E^A. As the deployment effort is a function of L^e and $M_{l_i^e}$, and L^B is reduced compared to L^A, the deployment effort in the target environment is reduced compared to the deployment effort in the source environment.

3.3 LSTM regression

The radio map can not contain all fingerprints for all locations in an environment. Recording data for every point would require too much data to capture and process, making the localization unfeasible to deploy in practice. Instead, the algorithm should check which coordinates in the radio map are the nearest and interpolate between those. The variability of RSSI over time makes this process more challenging. It turns out that an LSTM model is good at such a problem.
Xu et al. found that the sequence of RSSI is temporally correlative [12]. Therefore we capture a sequence of RSSI

values per location. As LSTM is particularly good for sequence problems, this Recurrent Neural Network is used in this research.

The model for environment A, which we call *LSTM* in this research, consists of a normalization layer, two LSTM layers, and three Dense layers. The structure is shown in figure 2. The normalization layer centers all input values around 0, with a standard deviation of 1. The LSTM layers both contain 100 cells. Then a dense layer with 100 cells and a dense layer with 50 cells are added. Experiments indicated that these amounts of cells provide the highest accuracy on our given problem. Finally, the last Dense layer contains two cells, such that both the x and y coordinate are output.

Figure 3 shows a visualization of the three-dimensional input of the model. The first dimension represents the number of samples. The more data available, the larger this dimension will be. For environment A , the first dimension will be bigger than for environment B, as more data is recorded in environment A.

The second dimension represents time-series. Even though multiple measurements per location are obtained in sequence, experiments showed that setting the second dimension to 1 instead of $M_{l_i^e}$, gave better accuracy.

The size of the third dimension represents the number of features. As $N^A \neq N^B$, the number of features is set to the biggest of the two environments. The data set with the least amount of features is padded with columns with only the minimal value (-110.0 dB). In figure 3 the data set with N^B features, which is the yellow part, is padded, with the brown part, such that the 3rd dimension of the data set of environment B matches the 3rd dimension of the data set of environment A.Adding a number of those padding columns did not affect the performance. However, when half the amount of features are added as padded columns, and these columns were randomly shifted, the accuracy dropped significantly. In that case, the effective-non-padded - features of the target data set do not line up with the useful features of the source data set. Certain features of the source data set will be unused, as they are mapped to padded columns in the target data set. In addition, certain features of the target data set cannot use the trained features of the source data set, as they were padded columns, which do not provide useful insights into the environment. In this research $N^A > N^B$, which is a valid assumption for other Transfer Learning problems, as the data set of the source environment is way bigger than the data set of the target environment. In this case, features of the target data set will line up with features of the source data set, and padding columns will not cause lower accuracy.

The model in this research is trained using the Adam optimizer, with a default learning rate of 0.001. In the fine-tuning step, which is explained in section 3.4, a learning rate of 0.0001 is used instead. The mean absolute error is the loss function. Training and validation loss are compared while training the model to prevent overfitting. The validation split is 20%. Different amounts of data require a different amount of training epochs. Therefore, the amount of epochs varies per experiment.

3.4 Transfer learning with LSTM

Figure 3 shows the architecture of Transfer Learning. The pink parts of the diagram are for environment A. The yellow parts are for environment B. The model of environment A, as shown in figure 2, is trained with a large data set. From this model, the final three dense layers are removed, and the LSTM layers are frozen. This model is the base part of our model. A new layer is added, such that we have a model for environment B, which is called *TL* in this research. This model for the target environment is trained on a small data set. After this training, the whole model for environment B, including the frozen LSTM layers, is unfrozen. The model trains again on the target data set, with a small learning rate. This step is called fine-tuning. The model after this fine-tuning step is called *TL+FT* in this research.

The degree to which the model represents the physical environment determines the accuracy with which the model can predict locations based on its input. The Transfer Learning architecture supports the model in learning the characteristics of wireless signal propagation. In the source environment, pink in figure 1, much data is available. Therefore, the model represents environment A well. This model is not a good representation of environment B, as that environment has different characteristics. There is another number of sending nodes, and those nodes are at other coordinates. Walls and furniture, which influence wireless signal propagation, are at different locations as well. These features are high-level, meaning that they are specific to an environment. There are more abstract features of the environment, that are shared between different environments. These features are called low-level.

Levels of abstraction are also present in machine learning models. The first layers represent low-level features, and the higher layers represent high-level features. The prediction layer, the last layer of the model, is the most specific to the environment, as it outputs coordinates that only make sense in that environment. Only the low-level representation of environment A is kept, as the higher layers are removed. The LSTM layers that are kept are frozen, to ensure that the knowledge is not overwritten while training on environment B. The newly added top layers can learn to represent high-level characteristics of the new environment, by using the low-level characteristics of the old environment. The low-level characteristics are helpful, as only limited data is available in environment B. The low-level representation does not map one-to-one on the new environment, so at last, the whole model is fine-tuned.

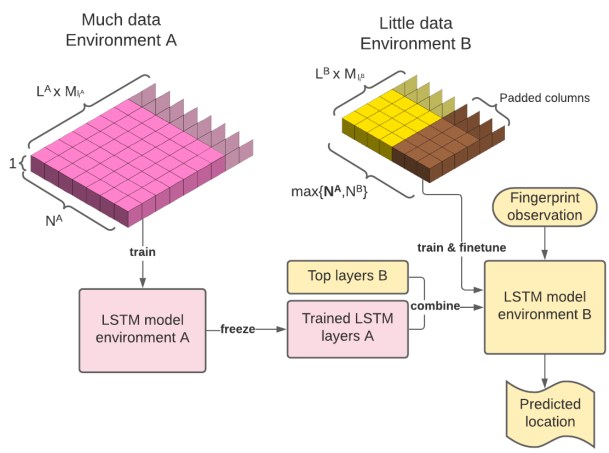


Figure 3: Architecture of the Transfer Learning process

4. EXPERIMENTS

In this section, the experiments to validate the accuracy of the proposed method are described. The setup of two experiments is explained first, after which the results of both experiments are analyzed.

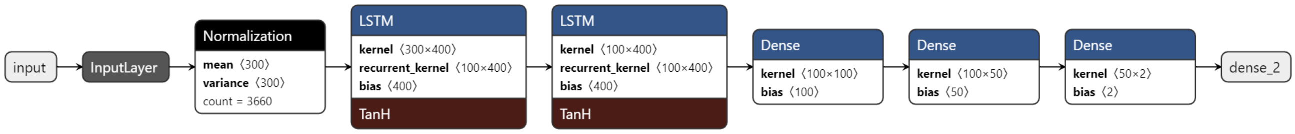


Figure 2: The LSTM model

4.1 Experimental setup

4.1.1 *Data collection*

Data for this research is collected in two buildings of the University of Twente. The first building is the Designlab, of which the floorplan is shown in figure 4. The floorplan is the same as used in the research of Le et al. [3]. The Designlab is the source environment (A). Note that the shown distribution of Bluetooth beacons is outdated, since not all beacons are active anymore, and some are moved. Since this research does not use the location of sending nodes, this can be ignored.

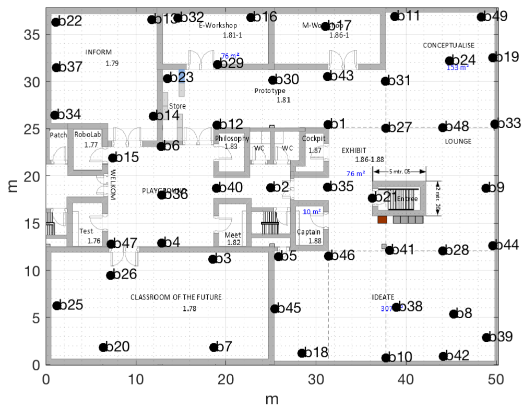


Figure 4: Environment A, the Designlab building, with the (outdated) distribution of Bluetooth beacons.

The second building is the Ravelijn, of which the floorplan is shown in figure 5. This map is taken from Google Maps and represents the target environment (B).
Both environments contain WiFi APs and Bluetooth beacons, which were already deployed. One should note, as mentioned in [3], that the sending nodes are deployed to provide the best signal coverage, and that the placement is not necessarily optimal for WiFi-based localization. For environment A the locations of these sending nodes are displayed as an example. Since the positions of the sending nodes are not needed in this research, they are not displayed in environment B.

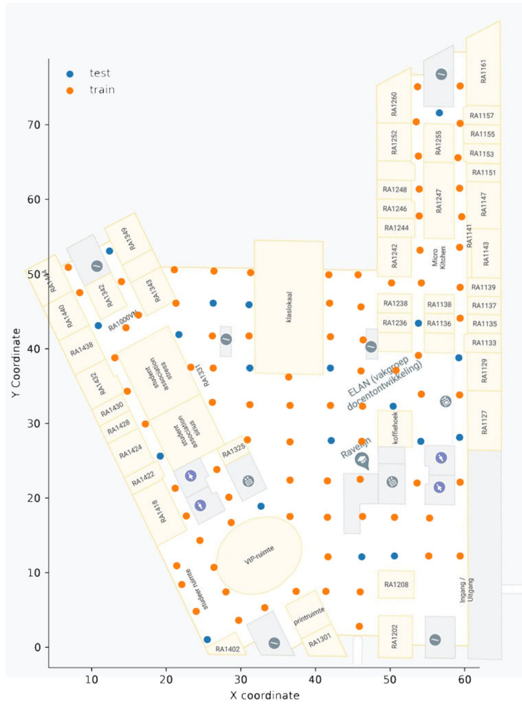


Figure 5: Environment B, the Ravelijn building with data point locations, randomly distributed in a test (blue) and a training (orange) data set.

An Android application has been developed for collecting RSSI data. The exact location on the floorplan can be indicated and measurement for that location can be started. A single measurement requests a WiFi scan and scans for Bluetooth signals for 2 seconds continuously. After both WiFi and Bluetooth data is retrieved, the application writes this data to a CSV file. At every location, 15 measurements are taken, which takes about 30 seconds per location. Sending nodes that are not received in the current scan default to the minimal value (-110dB).
In environment A, measurements at 152 different locations are taken. Since two phones are used for data collection, for every location 30 measurements are taken. In environment B, measurements at 102 different locations are taken, with 15 measurements each. Different models are trained with subsets of this data, of which the results are explained in section 4.2.

4.1.2 *Description of experiments*

Experiment 1 In environment A, the influence of the amount of data on the prediction accuracy is examined. Different numbers of measurement locations will be used to train the model. The model will also be trained on all measurement locations, providing a baseline accuracy. It is expected that the more data is trained on, the higher the accuracy will be.
Experiment 2 The next experiment takes various amounts of measurement locations in environment B and compares

three models. The first model, *LSTM*, is an untrained model as described in figure 2, which is trained on the data. This model does not apply Transfer Learning techniques. It is expected that the accuracy is comparable to the model in experiment 1. The second model, *TL*, will be the model trained on environment A, with frozen layers and new top layers. This architecture is as described in 3.4, without the fine-tuning step. The third model, *TL+FT*, continues the *TL* model with additional fine-tuning applied. It is expected that the third model will perform better than the second model. It is also expected that the third model will perform better than the first model, although with a lot of data available the difference might not be very significant.

4.2 Experimental results

For all accuracy measurements, the data set is randomly shuffled by location. To account for the random nature of RNNs, the model is reset, trained, and evaluated five times for each different configuration. This section reports the averages of these results.
The data set for an experiment is split into a training and a test set. The ratio between these sets defines how much data is used for training. In other words, all data that is not used for training is used for testing. This splitting of the data set can be done in several ways. The data can be randomly divided according to the ratio, or the data can be grouped per location and then randomly divided according to the ratio. For both experiments, the latter option is used. If data has to be collected, it is more efficient to record more data at fewer locations than to record fewer data at more locations, since it takes a certain amount of time to move to the next measurement location.
To train the base model for the second experiment, the first approach of splitting the data set into a training and test set is used. By using this method, the data set is as diverse as possible. The accuracy is better if the model is trained at more locations with less data, then if the model is trained at less locations with more data.

4.2.1 Experiment 1

For this experiment, 4560 samples are collected in the Designlab building of the University of Twente. This data represents 152 different locations, with 30 measurements each. Table 1 shows the accuracy of this experiment. As expected, the accuracy is higher when the model trains on more data. The baseline accuracy of our model is 3.3 meters since this is the best accuracy obtained.
Sahar et al. found an accuracy of 2 meters for their LSTM architecture [8]. They explain that deep neural networks are very sensitive to hyper-parameter tuning. Our research spent little time finding the best hyperparameters, which might explain the difference in the accuracy. The focus of this research is on Transfer Learning, not solely on using the best LSTM-based localization. Chen et al. also show that improving the LSTM architecture results in a higher accuracy. They use feature extraction to obtain an accuracy of 1.75 meter [1].

Training locations	**122**	**92**	**61**	**31**	**15**
LSTM accuracy (m)	3.3	3.7	3.9	4.2	5.2

Table 1: Accuracy (m) of the LSTM model for various numbers of training data locations in the source environment

4.2.2 Experiment 2

For this experiment, 1530 samples are collected in the Ravelijn building of the University of Twente. This data is collected at 102 different locations, with 15 measurements each.
The result of the second experiment can be found in table 2 and figure 7. The table shows accuracies for the three models trained on several numbers of training locations. The graphs plotted in figure 7 show more insight into the distribution of these errors. These plots show the percentage of errors that are within a range in meters. For example, figure 7a shows that about 50% of all tests predictions have an error of 5 meters or less for both *LSTM* and *TL+FT*. However, the worst 20% of predictions for *LSTM* are worse than the worse 20% of predictions of *TL+FT*. It is noticeable that for every amount of training locations this is the case.

Training locations	**82**	**61**	**41**	**20**	**10**
LSTM accuracy (m)	4.8	5.4	6.3	11.6	17.1
TL accuracy (m)	6.9	7.5	8.4	8.7	13.1
TL+FT accuracy (m)	4.5	5	5.5	6.1	9.5

Table 2: Accuracy (m) of the second experiment for various numbers of training data locations in the target environment

Let us first take a look at the *LSTM* model that is trained on data in the target environment. As the amount of training locations decreases, the accuracy of this model significantly decreases as well. For low amounts of data points, the model has not enough data to learn environmental characteristics, so it can never make good predictions. One should notice the difference in accuracy between this model, and the model of experiment 1 (see table 1). The number of training locations does not match, but the accuracy of the same type of model is less for all amounts of training locations than in experiment 1. In experiment one, every data location had 30 measurements. In other words, for the same amount of training locations, the model had twice as much data. A model can learn better if more data is available, which explains the higher accuracy.
The accuracy of the model with some fine-tuning applied (*TL+FT*) is significantly better than the transferred model without fine-tuning (*TL*). This final transfer learning model, with fine-tuning, has higher accuracy than the LSTM model that is solely trained on the target environment data set. Not only the average error is lower, but the cumulative distribution shows that there are fewer large errors of more than 10 meters. In other words, to obtain the same accuracy, less data is needed for the model that uses Transfer Learning techniques compared to a basic LSTM model. To visualize this result, figure 6 shows 10 random test locations, as well as the corresponding LSTM predictions and the TL+FT predictions. This figure shows the case where 41 data locations in environment B are used to train.

5. CONCLUSION & FURTHER RESEARCH

In this paper, we propose an LSTM-based fingerprinting architecture with Transfer Learning techniques. By training an LSTM model on a source environment data set and applying Transfer Learning techniques, we have reduced the amount of data required for the target environment. We have developed a prototype in the form of an Android

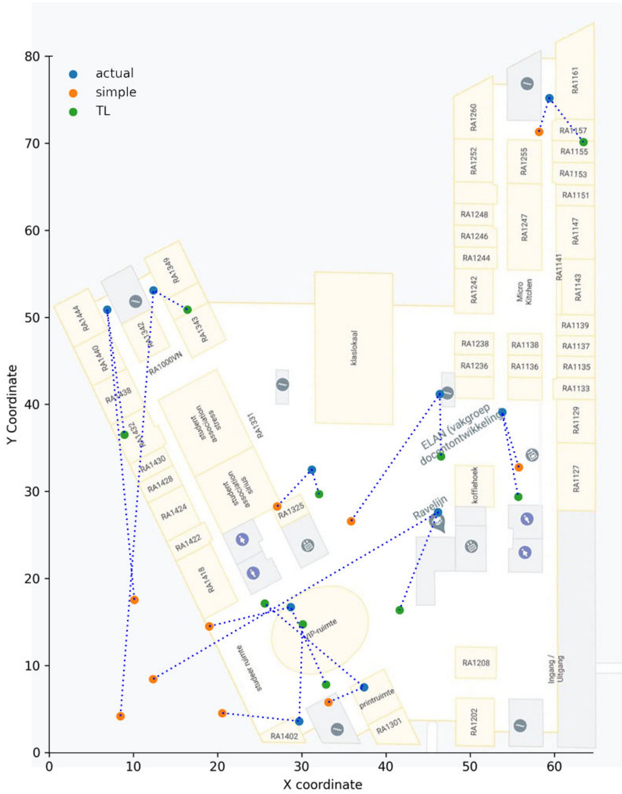


Figure 6: Environment B, with 10 random test locations (blue) and the *LSTM* prediction (orange) and *TL+FT* prediction (green), trained on 41 data locations.

application and have carried out experiments to evaluate the performance of our model. The accuracy of the model with Transfer Learning techniques applied is higher than the model that did not use Transfer Learning. The goal of this research is to reduce deployment costs. Since accuracy is correlated to the amount of data used for training, we can conclude that deployment costs have decreased because of TL techniques.
Assuming a particular application of indoor localization requires accuracy of X meters needs an amount of A data to train on. A pre-trained model would only require an amount B of data to get this accuracy. This research shows that B is smaller than A. In other words, less data is required for this application when a pre-trained model with TL techniques is used. The effort takes to collect data mostly defines the deployment costs. Since less data is required, the deployment costs decreased.

This experiment has been carried out in two buildings of the University of Twente. Those buildings have the same WiFi and Bluetooth infrastructure. Future research could be done on the performance of Transfer Learning for LSTM-based indoor localization in more diverse environments.
This research uses two-dimensional regression, providing x and y coordinates. Applications might benefit from a third dimension, for instance, the floor level. The experiments of this research could be repeated while taking into account the z dimension.
During this research, buildings were only partially accessible due to the Covid-19 pandemic. Therefore, it was not possible to take data measurements at every desired location in the environment. The distribution of data locations is not uniform and might affect performance. The experiment environments were static, in comparison to buildings in normal daily use. Due to university regulations, furniture was at pre-defined places and not moved often. People stayed at their locations most of the time and did not walk around a lot. There were also fewer people in the building than usual. All these factors might impact the performance of the localization.

References

[1] Z. Chen, H. Zou, J. F. Yang, H. Jiang, and L. Xie. WiFi Fingerprinting Indoor Localization Using Local Feature-Based Deep LSTM. *IEEE Systems Journal*, 14(2):3001–3010, jun 2020.

[2] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, oct 2017.

[3] D. V. Le and P. J. Havinga. SoLoc: Self-organizing indoor localization for unstructured and dynamic environments. In *2017 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2017*, volume 2017-Janua, pages 1–8. Institute of Electrical and Electronics Engineers Inc., nov 2017.

[4] M. Nowicki and J. Wietrzykowski. Low-effort place recognition with WiFi fingerprints using deep learning. nov 2016.

[5] S. J. Pan, D. Shen, Q. Yang, and J. T. Kwok. Transferring Localization Models Across Space. Technical report.

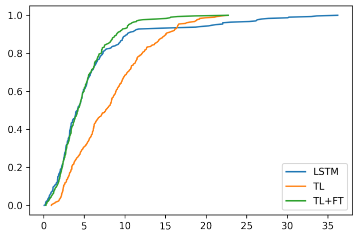
[6] S. J. Pan and Q. Yang. A survey on transfer learning, 2010.

[7] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training Recurrent Neural Networks. *30th International Conference on Machine Learning, ICML 2013*, (PART 3):2347–2355, nov 2012.

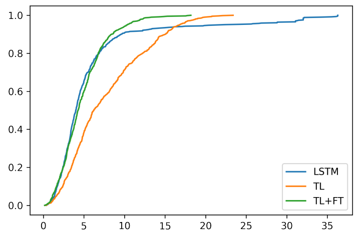
[8] A. Sahar and D. Han. An LSTM-based indoor positioning method using Wi-Fi signals. In *ACM International Conference Proceeding Series*. Association for Computing Machinery, aug 2018.

[9] X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L. L. Chen, J. Qin, and Z. Wang. Cnnloc: Deep-learning based indoor localization with wifi fingerprinting. In *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, pages 589–595. Institute of Electrical and Electronics Engineers Inc., aug 2019.

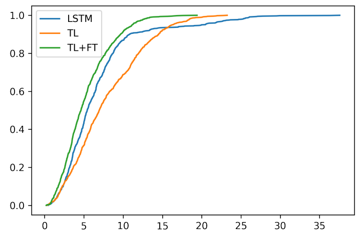
[10] S. Sorour, Y. Lostanlen, S. Valaee, and K. Majeed. Joint Indoor Localization and Radio Map Construction with Limited Deployment Load. *IEEE Transactions on Mobile Computing*, 14(5):1031–1043, may 2015.



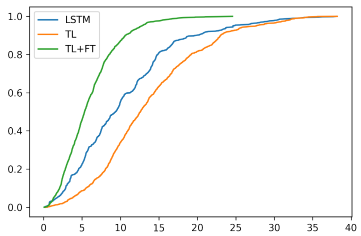
(a) 88 training locations



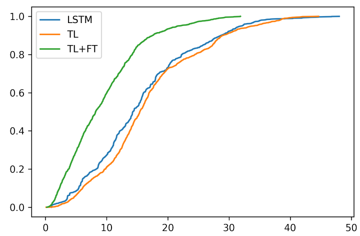
(b) 61 training locations



(c) 41 training locations



(d) 20 training locations



(e) 10 training locations

Figure 7: Cumulative distribution of accuracy (m) of the second experiment for various numbers of training data locations in environment B

[11] M. Y. Umair, K. V. Ramana, and Y. Dongkai. An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN. In *Proceedings - 2014 IEEE Computers, Communications and IT Applications Conference, ComComAp 2014*, pages 19–23. Institute of Electrical and Electronics Engineers Inc., jan 2014.

[12] B. O. Xu, X. Zhu, and H. Zhu. An Efficient Indoor Localization Method Based on the Long Short-Term Memory Recurrent Neuron Network.

[13] Q. Yang, S. J. Pan, V. W. Zheng, H. Kashima, S. Suzuki, S. Hido, Y. Tsuboi, T. Takahashi, T. Idé, R. Takahashi, A. Tajima, Y. Katori, Y. Qu, C. Li, X. Z. Wang, F. Guo, X. Gao, Z. Sun, J. Qi, J. Liu, and Y. Chen. Estimating location using Wi-Fi. *IEEE Intelligent Systems*, 23(1):8–9, jan 2008.

[14] S. Yiu, M. Dashti, H. Claussen, and F. Perez-Cruz. Wireless RSSI fingerprinting localization, feb 2017.

[15] M. A. Youssef, A. Agrawala, and A. U. Shankar. WLAN location determination via clustering and probability distributions. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications, PerCom 2003*, pages 143–150, 2003.

[16] X. Zhang, Y. Mei, H. Jin, and D. Liang. TL-FCMA: Indoor Localization by Integrating Fuzzy Clustering with Transfer Learning. In *Proceedings of 2018 6th IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2018*, pages 372–377. Institute of Electrical and Electronics Engineers Inc., nov 2018.

[17] V. W. Zheng, E. W. Xiang, Q. Yang, and D. Shen. Transferring Localization Models Over Time *. Technical report.