Annals of Telecommunications (2022) 77:345–357
https://doi.org/10.1007/s12243-021-00853-z

Generalization aspect of accurate machine learning models for CSI-based localization

Abdallah Sobehy[1] · Éric Renault[2] · Paul Mühlethaler[3]

Received: 14 October 2020 / Accepted: 4 May 2021 / Published online: 14 June 2021

Abstract

Localization is the process of determining the position of an entity in a given coordinate system. Due to its wide range of applications (e.g. autonomous driving, Internet-of-Things), it has gained much focus from the industry and academia. Channel State Information (CSI) has overtaken Received Signal Strength Indicator (RSSI) to achieve localization given its temporal stability and rich information. In this paper, we extend our previous work by combining classical and deep learning methods in an attempt to improve the localization accuracy using CSI. We then test the generalization aspect of both approaches in different environments by splitting the training and test sets such that their intersection is reduced when compared with uniform random splitting. The deep learning approach is a Multi Layer Perceptron Neural Network (MLP NN) and the classical machine learning method is based on K-nearest neighbors (KNN). The estimation results of both approaches outperform state-of-the-art performance on the same dataset. We illustrate that while the accuracy of both approaches deteriorates when tested for generalization, deep learning exhibits a higher potential to perform better beyond the training set. This conclusion supports recent state-of-the-art attempts to understand the behaviour of deep learning models.

Keywords Indoor localization · Channel state information · MIMO · Deep learning · Neural networks · KNN · Ensemble learning · Generalization

1 Introduction

The location of entities is generally determined by 2D or 3D coordinates in some coordinate system. This knowledge serves a wide range of applications such as autonomous driving, routing, environmental surveillance, etc. Localization solutions vary depending on several factors including the available sensors (e.g. cameras, LiDARs, GPS) or the environment (e.g. indoors or outdoors). Each context introduces some constraints that require a solution that takes them into account. For example, in an outdoor context, GPS is widely used for localization, whereas in indoor context, GPS service is not reachable and thus some other forms of information are needed to compensate for the absence of GPS.

One family of localization methods is known as range-based localization. In this method, a physical phenomenon is used to estimate the distance between nodes. Then, the relative positions of nodes within a network can be computed geometrically [1]. One of the most used phenomena is the Received Signal Strength Indicator (RSSI). RSSI is an indication of the received signal power. It is mainly used to compute the distance between a transmitter and a receiver since the signal strength decreases as the distance increases. In [2], the distances between nodes along with the position information of a subset of nodes, known as the anchor nodes, are used to locate other nodes in a Mobile Adhoc Network (MANET). This is achieved using a variant of the geometric triangulation method. The upside of RSSI is that it does not need an extra hardware to be computed and is readily available. Another physical measure to compute the

✉ Abdallah Sobehy
a.sobehy@gmail.com

Éric Renault
eric.renault@esiee.fr

Paul Mühlethaler
Paul.Muhlethaler@inria.fr

[1] Samovar, CNRS, Télécom SudParis, 9 Rue Charles Fourier, 91000 Évry, France

[2] EISEE Paris, 2 Boulevard Blaise Pascal, 93160 Noisy-le-Grand, France

[3] Inria, 2 Rue Simone IFF, 75012, Paris, France

distance between devices is the Time-Of-Arrival (TOA) or Time-Difference-Of-Arrival (TDOA). Here, the time taken by the signal to reach the receiver is used to estimate the distance between devices. Using TOA in localization proves to be more accurate than RSSI, but requires external hardware to synchronize nodes [3]. In the case where only the distance information is available, a minimum of three anchor nodes with previously known positions are needed to localize other nodes with unknown positions.

Due to the sensitivity of RSSI to multipath fading and environmental noise [4], the localization trend shifted towards a more stable form of information known as Channel State Information (CSI). With the use of Multiple Input Multiple Output (MIMO) antennas and orthogonal frequency-division multiplexing (OFDM) each antenna receives multiple signals on adjacent subcarriers at which CSI is computed. CSI contains richer and more stable information than RSSI. Thus, it is more suitable for accurate finger-printing-based localization.

CSI represents the change that occurs to the signal as it passes through the channel between the transmitter and the receiver, e.g. fading, scattering, and power loss [5]. Equation 1 specifies the relation between the transmitted signal $T_{i,j}$ and the received signal $R_{i,j}$ at the ith antenna and the jth subcarrier. The transmitted signal is affected by both the white noise N and the channel which is represented by the complex number $CSI_{i,j}$.

$$R_{i,j} = T_{i,j} \cdot CSI_{i,j} + N \tag{1}$$

In the following section, we discuss and compare different state-of-the-art machine learning models used to solve the localization problem. We compare their structure and how the model's architecture affects the estimation results. In Section 3, we briefly present our MLP NN-based solution from our previous work [6] which we are extending here in this publication. Then, the MLP NN predictions are combined with that of the KNN-based solution [7] using ensemble learning. In Section 4, we evaluate the accuracy of the presented learning techniques in indoor and outdoor environments and introduce two methods of test set selection to evaluate the generalization aspect. Section 5 includes the conclusion and our insights for possible future work to enhance the generalization aspect of the learning process.

2 Related work

With the availability of access to cheap data and cheap computational resources, machine learning methods have been intensively used in a plethora of applications [8]. Deep learning in particular has emerged in recent years as the dominant solution to multiple problems, e.g. image recognition [9] and natural language processing [10]. The principal evaluation criterion of machine learning models is the value of error. The error can be the percentage of correct predictions in classification contexts or the difference between prediction and ground truths in continuous contexts. In the localization context, the error is commonly chosen to be the root mean square error (RMSE) between the estimated position and the ground truth position in localization context. We argue that the model's ability to generalize beyond the training data must be considered alongside the absolute error value for performance evaluation.

One classical machine learning model is K-Nearest Neighbor (KNN). In [7], KNN has been used to localize a transmitter based on the magnitude component of CSI calculated at a 2 × 8 MIMO antenna. KNN requires a choice of neighboring criteria and the value k which is the number of neighbors used in estimating the position. The Euclidean distance is chosen as the neighboring criterion and the value of k is set to one. This method achieves a 2.3-cm RMSE which is, to the best of our knowledge, the lowest error on the dataset provided in the indoor positioning competition prepared by IEEE's Communication theory workshop [11]. This result is interesting since the classical KNN approach outperformed the MLP NN method [12] tested on the same dataset achieving an error of 4.5 cm. Even when ensemble learning was used by combining predictions of multiple different MLP NNs [6], the KNN method still achieved a better mean error.

KNN method outperformed deep learning methods from a localization error perspective. We are interested in evaluating the ability of the deep learning model to generalize which can give it a decisive edge over the KNN method despite the higher error localization error. An impressive indoor localization accuracy has been achieved using a larger MIMO antenna (8 × 8) and a very deep Convolutional Neural Network (CNN) [13]. The proposed CNN architecture is based on the DenseNet architecture [14] which was originally created for the well-known image recognition competition ImageNet [15]. The DenseNet [14] proved itself as one of the best solutions in terms of classification accuracy. The CNN-based solution [13] is tested on a dataset different from the dataset on which our methods are tested. However, we are not aware of a CSI-based indoor localization solution that is able to achieve an error lower than the 17 mm error of [13].

The generalization aspect was tested in [13] where the highly accurate CNN was trained on data collected in a room with Line-of-sight (LoS) transmissions. The test set was made up of LoS CSI readings using the same MIMO antenna, and the same distance to transmitter but in a different room. Even though the surrounding conditions are similar (same antenna and LoS transmissions), the CNN completely fails to predict the position of the transmitter in the new room. More precisely, the error jumps from 17 mm to ≈ 700 mm. This result is quite surprising considering the fact that the localization area is less than 8 m^2. The authors conclude that while the LoS component should be the same in both rooms, the multipath component is not the same because of reflection against different set of objects. Thus, their take away is that a good model needs to learn multi-path components of the environment to be able to perform well. While this conclusion sounds legit, we believe there might be other reasons for the failure of the learning model. The experiments performed in this work makes us lean towards believing that the failure could be related to the learning model's architecture, and other solutions might open the door towards generalizing to different environments.

3 Methodology

3.1 Experimental setup

The experiment on which our solution is tested is carried out by the authors and organizers of the IEEE CTW indoor positioning competition [16]. In [11], the authors present the MIMO antenna used in the competition. They carry out an experiment where a robot carrying a transmitter traverses a 4 × 2 meter table and communicates with the 8 × 2 MIMO antenna. The transmission frequency is 1.25 GHz and the bandwidth is 20 MHz. Signals are received at each of the 16 subantennas over 1024 subcarriers from which 10 % are used as guard bands. Using a convolutional neural network (CNN), the authors used the real and imaginary components of the CSI as input to the learning model to estimate the robot's position. The authors publish the CSI readings and the corresponding positions (≈ 17,000 samples) readings which are used as the test bed for our algorithm. Figure 1 demonstrates the experimental setup as well as a sketch of the MIMO antenna and the position of its center. The lower part of the figure shows the table which is traversed through the experiment and the MIMO antenna. The upper part shows a sketch of the MIMO antenna displaying its center at (3.5, − 3.15, 1.8)m in the local coordinate system. The distance between adjacent antennas is $lambda/2$ which is computed from the carrier frequency.

The CSI can be represented in polar or cartesian coordinates. Thus, the learning model input can be magnitude, phase, real, or imaginary or a combination of these components. The magnitude was found to be statistically stable with respect to time. In other words, transmitting from the same position at different times yields very similar magnitude components. On the other hand, the values vary strongly with phase, real, and imaginary components [7, 12, 18]. Thus, the magnitude component is chosen as the input feature to the MLP NN model.

In order to reduce the noise and dimensionality of magnitude values, the magnitude values at each antenna are divided into 4 subdivisions. Polynomial regression is then performed with different degrees to predict a polynomial line that accurately presents each of the four regions. Figure 2 shows the magnitude values at one of the 16 antennas and the 4 polynomial lines approximating the values. The four lines are concatenated and smoothed at the borders between adjacent lines using a weighted averaging method. More information on the fitting process is detailed in [12]. Instead of using all points along the fitted line to represent the magnitude component, a reduced number of equidistant values are selected based on the chosen learning model. The number of values is selected empirically aiming to reduce the input's dimension without affecting the stability of the learning model's accuracy.

3.2 MLP NN ensemble method

The input to the MLP NN model is chosen empirically to be 66 equidistant points along the fitted line. In ensemble learning, multiple learning models are trained and their predictions are combined to yield better results. We built different learning models by making changes to the input format, hyperparameters, and training input samples. As for the input format, some models train on magnitude values, while other models train on the differences between consecutive magnitude values along the fitted lines. These differences represent the slope values between two points along the line. Hyperparameter values are manually varied creating different learning models. Table 1 summarizes some of the hyperparameters and the attempted ranges to train different learning models.

Data augmentation is also used to generate more training examples and to enhance the model's ability to learn. This is achieved by jiggling the magnitude values and the predicted positions. More details on the data augmentation process are found in [6].

We chose the most accurate 19 learning models, and their predictions are averaged using different methods to predict the location of the transmitter. The tested averaging methods are:

1. Mean: The simplest way to mix the results is to compute the arithmetic mean position of all the predictions.
2. Weighted mean: Each of the MLPs is given a weight that is proportional to its individual localization accuracy. Thus, predictions from more accurate models are given higher weights. The final prediction is a weighted average of the individual predictions.
3. Weighted power mean: The impact of weights is further magnified by raising them to a certain power before computing the weighted average.

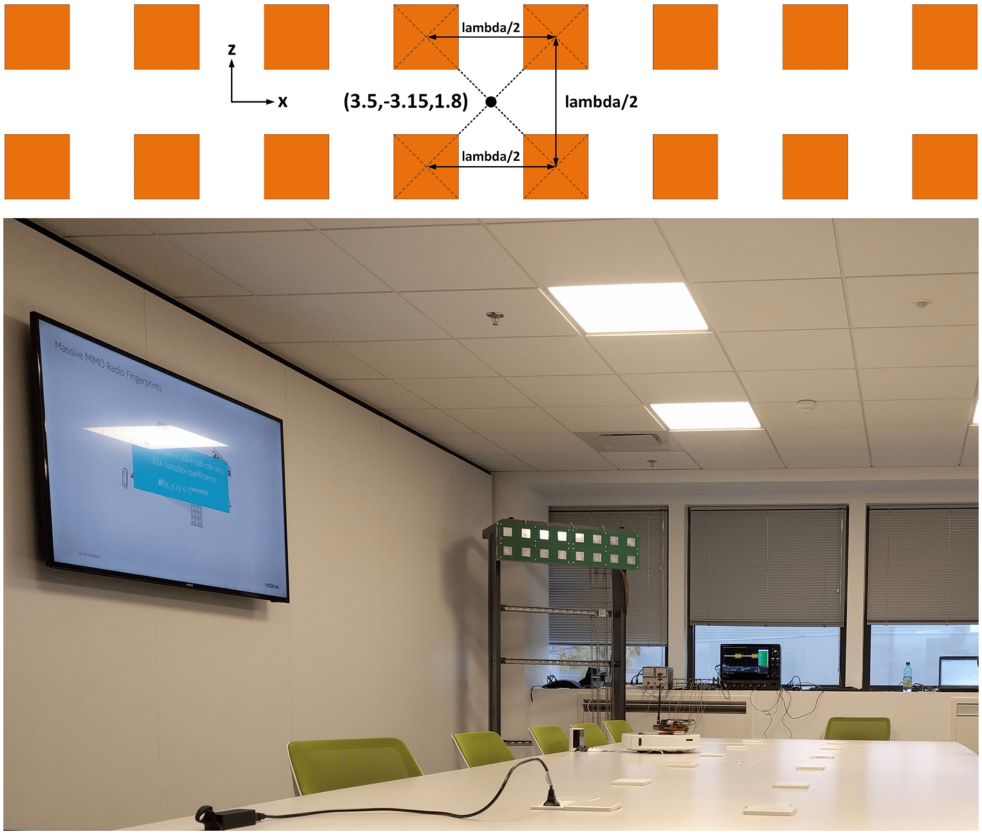


Fig. 1 Experimental setup (bottom) and a sketch of the MIMO antenna (top) [11]

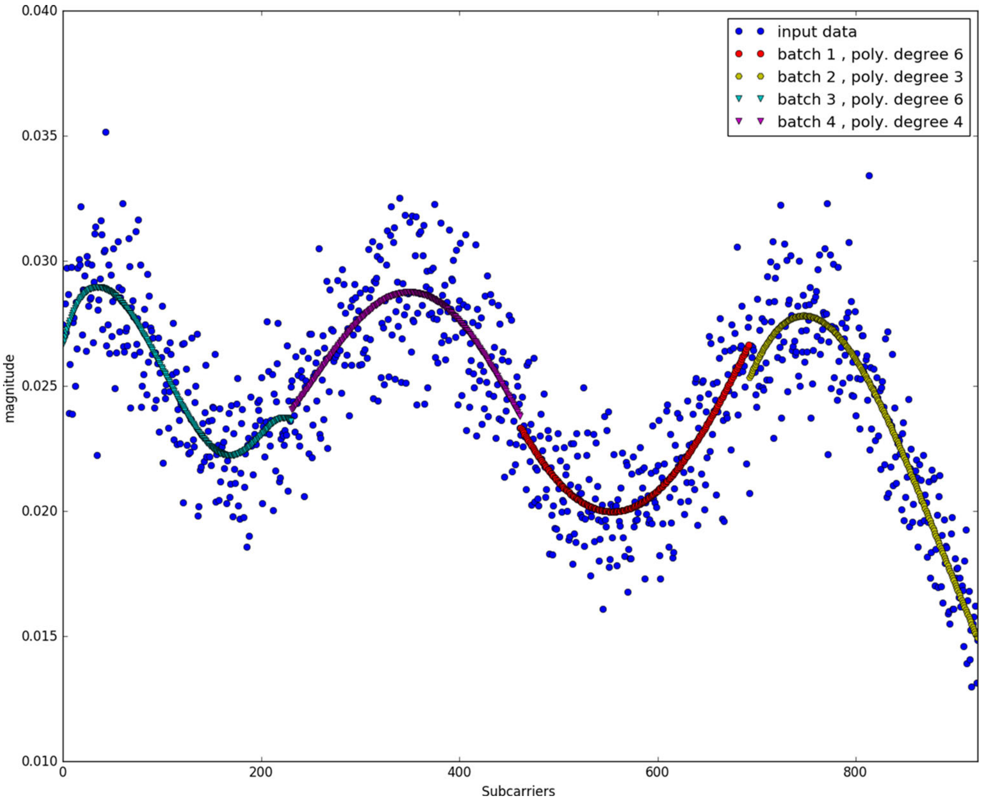


Fig. 2 Fitting a line through the magnitude values [6]

Table 1 Hyperparameters ranges

Hyperparameter	Range
Number of layers	[4,5,6,7,8]
Units per layer	[128,256,512,1024,1200]
Epochs	[50,100,150,200]
Activation function	[relu, selu, tanh, softmax]
Learning rate	$[25 \times 10^{-5}, 5 \times 10^{-4}, 1 \times 10^{-3}]$
L2 regularization	$[\text{without}, 1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}]$
Dropout percentage	$[1\%, 2\%, \ldots, 10\%]$

4. Median: The idea is to pick one of the predictions that is closest to all other predictions. This makes sense when the ensemble has three or more MLPs. This mitigates the effect of the large errors of some predictions.
5. Random: The final prediction is a randomly selected individual prediction.
6. Best pick: This is used as an indication of the best possible result that could be attained from the individual predictions. The final prediction is the closest individual prediction to the actual position. This is not feasible in the normal test scenario where the actual positions are not known.

3.3 K-nearest neighbors method

With KNN [7], the estimation accuracy was found to be stable using only 33 equidistant points. The neighboring criterion is the Euclidean distance between CSI magnitude values which is defined in Eq. 2. A is the number of antennas, S is the number of selected equidistant points along the polynomial line, and M^1 and M^2 are two sets of $S \times A$ magnitude values for all 16 antennas.

$$dist_{M^1,M^2} = \frac{1}{A}\sum_{a=1}^{A}\sqrt{\sum_{n=1}^{S}\left(M^1_{a,n} - M^2_{a,n}\right)^2} \tag{2}$$

With k value chosen to be 1, for each test sample, the whole training set is traversed and the position of the training sample that is closest to the test sample according to Eq. 2 is chosen as the predicted position.

4 Evaluation

4.1 Accuracy evaluation

The dataset is split into 90% training set and 10% test sets. The validation set is provided by the IEEE CTW indoor positioning organizers on the competition day. The validation set is composed of 2000 CSI samples without the corresponding positions. Participants used their prepared models to predict the corresponding positions. The position predictions are sent by each team to the organizers for evaluation.

The best RMSE achieved by the MLP NN ensemble learning method [6] is ≈ 3.1 cm. This error was achieved using the most accurate 11 MLP NN models. While the best individual MLP NN achieves 3.9-cm RMSE, using data augmentation and combining its predictions with less accurate models yield better results.

This outcome encouraged us to combine the predictions of KNN with that of the MLP NN ensemble. KNN method [7] achieves a 2.3-cm RMSE which is, to the best of our knowledge, the lowest achieved error with this particular dataset. While the ensemble yields lower accuracy, it might be possible to further improve the accuracy when combining both methods together.

Figure 3 illustrates the result of using ensemble learning with MLP NNs and KNN methods. Three categories of results are depicted:

1. Deep MLP NNs estimations are combined using different combination methods: mean, median, best pick, and stochastic pick.
2. KNN method estimation error.
3. Averaging KNN predictions with the median predictions of the MLP NN ensemble.

The x-axis shows the number of MLP NNs used in the ensemble. The number between brackets is the individual RMSE of the added NN to the ensemble. To clarify, the first MLP NN has a 3.9-cm RMSE. As we move to the right, at each step, an MLP NN is added to the ensemble with the error between brackets. The stochastic pick yields the worst results as it randomly selects the prediction of one of the MLP NN models in the ensemble. The other extreme is the best pick which gives an intuition about the possible accuracy if there is a way that ensures selecting the best prediction among all models in the ensemble. The remaining combination methods (excluding KNN) show an improvement in localization accuracy when more MLP NNs are added to the ensemble. However, the accuracy starts to deteriorate when models with larger errors are added. The horizontal dashed line represents the use of KNN alone which yields an error of 2.3 cm. The KNN + median line is the averaging between the KNN prediction and the median combination of the MLP ensemble. It is clear that combining the predictions of KNN and the ensemble does not break new grounds in the estimation accuracy but rather yields an error between both paradigms.

4.2 Generalization evaluation

Combining the predictions of KNN and MLP NN ensemble did not improve accuracy. In this section, we compare the

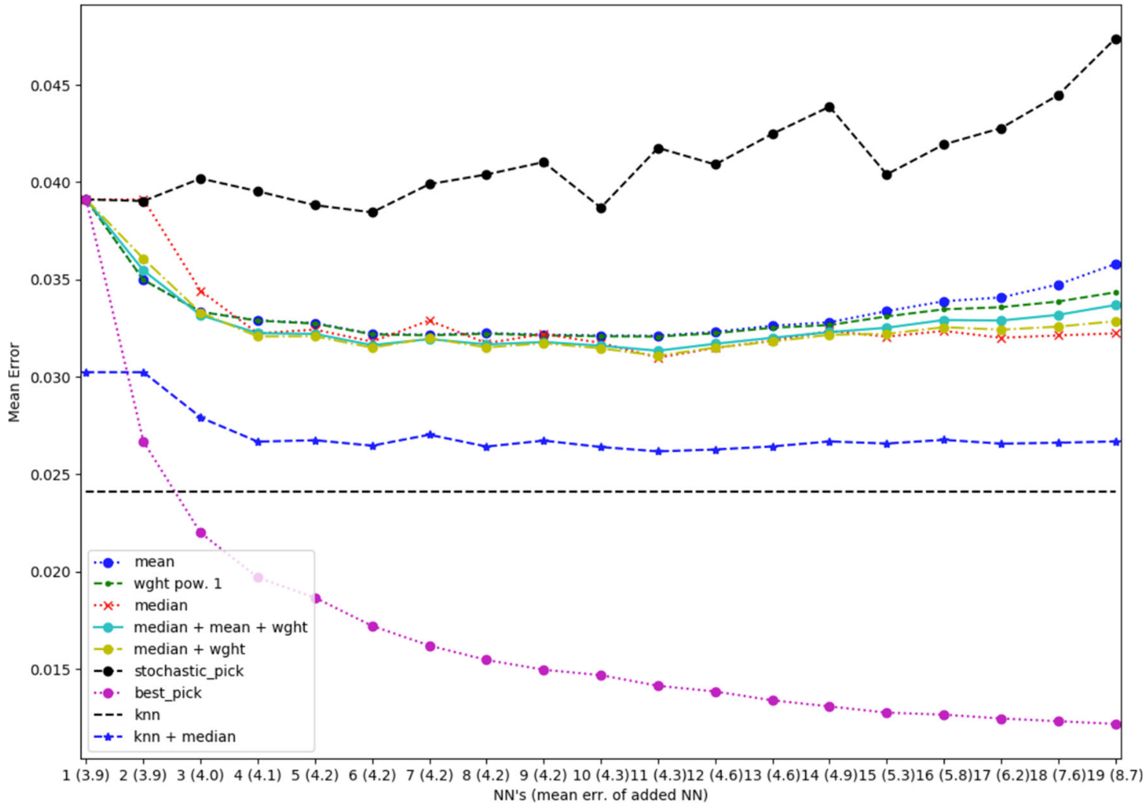


Fig. 3 Ensemble learning using MLP NNs and KNN

generalization ability of the KNN method against that of the MLP NN method. Since we focus on generalization rather than accuracy in this section, we use a light MLP NN model that achieves ≈ 6.5 cm error. The generalization capabilities of MLP NN and KNN are tested by altering the splitting technique of the train and test sets from the available dataset. Commonly, the splitting is achieved by uniform random selection of samples for the train and test sets. To test the generality of each model, we maintain the 90% and 10% sizes of train and test sets, respectively. However, the test set samples are selected such that no test set position intersect with any of the training set positions. In this case, a model with good generalization ability must be able to interpolate or extrapolate from the train set samples to predict sensible estimations for the test set samples. We propose two methods for test set selection, square and sequential selections that test the extrapolation and interpolation capabilities, respectively.

4.2.1 Square test set selection

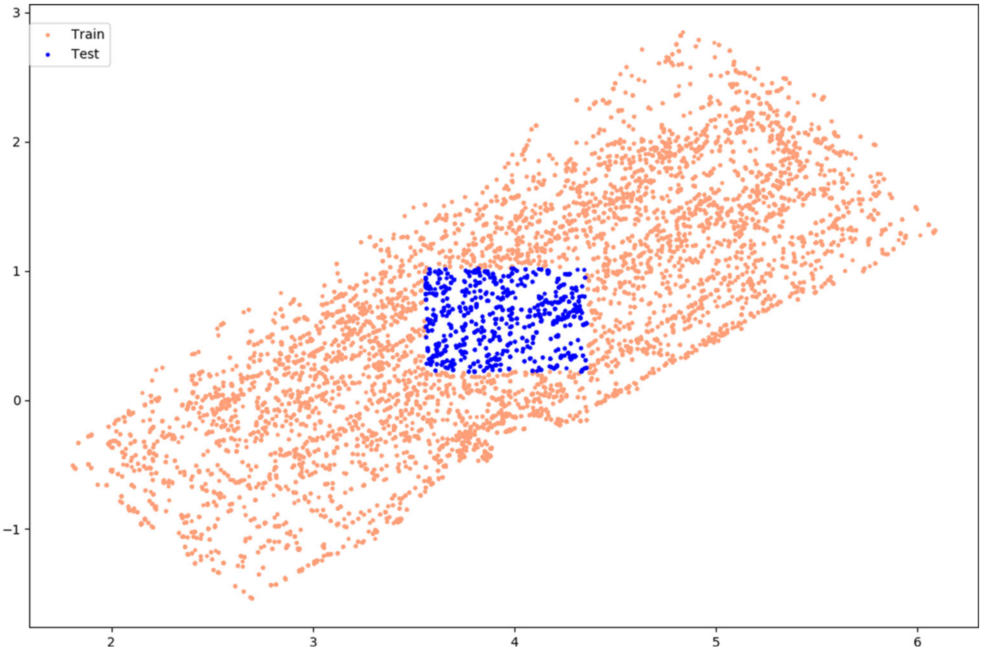
The test set samples in this selection technique are selected from a square in the middle of the table which the transmitter traverses. The size of the square is adjusted so that the size of the test is ≈ 10% of the whole dataset. The training set and test set are shown in Fig. 4.

The test sample square area makes the prediction task very challenging. The blue square region is like a blind spot which the learning model has to predict without having any experience (data) in it. This test set distribution examines the extrapolation capability of the algorithm as most of the test set samples are confined in a region where no training samples exist. Figure 5 shows KNN's predicted sample positions linked with edges to the corresponding ground truths. Figure 6 demonstrates the error distribution of the predicted positions.

It can be seen from Fig. 5 that all predicted positions are outside the test square region. This is expected because the value of k is one and, thus, the KNN finds the closest training sample which is always outside the test region. The error distribution presented in Fig. 6 does not show a particular known distribution. The mean error is 0.73m and the standard deviation is 0.53 m. This large errors indicates a complete failure of the model since the error is 0.023 m when the test set samples were drawn randomly.

The chosen MLP NN achieves a mean error of 0.065 m with random test set selection. Since the aim of this experiment is to test the generalization ability, there is no need to train a very complex NN. The difference between the error in the random test set and the square test set is enough to indicate the generalization ability. Figure 7 illustrates a map relating the predicted positions to the

Fig. 4 Square test set selection



ground truths. A clear difference between the estimations of the MLP NN and that of the KNN is that some of the estimations are inside the test set region. This can be explained by the fact that the MLP NN forms a function of the input without a restriction related to the train set positions. Thus, some of the test set samples yielded some estimations inside the test set region. The estimated positions still appear to be random. Figure 8 shows the error distribution of the MLP NN estimations. While the mean error and standard deviation are less than that of KNN,

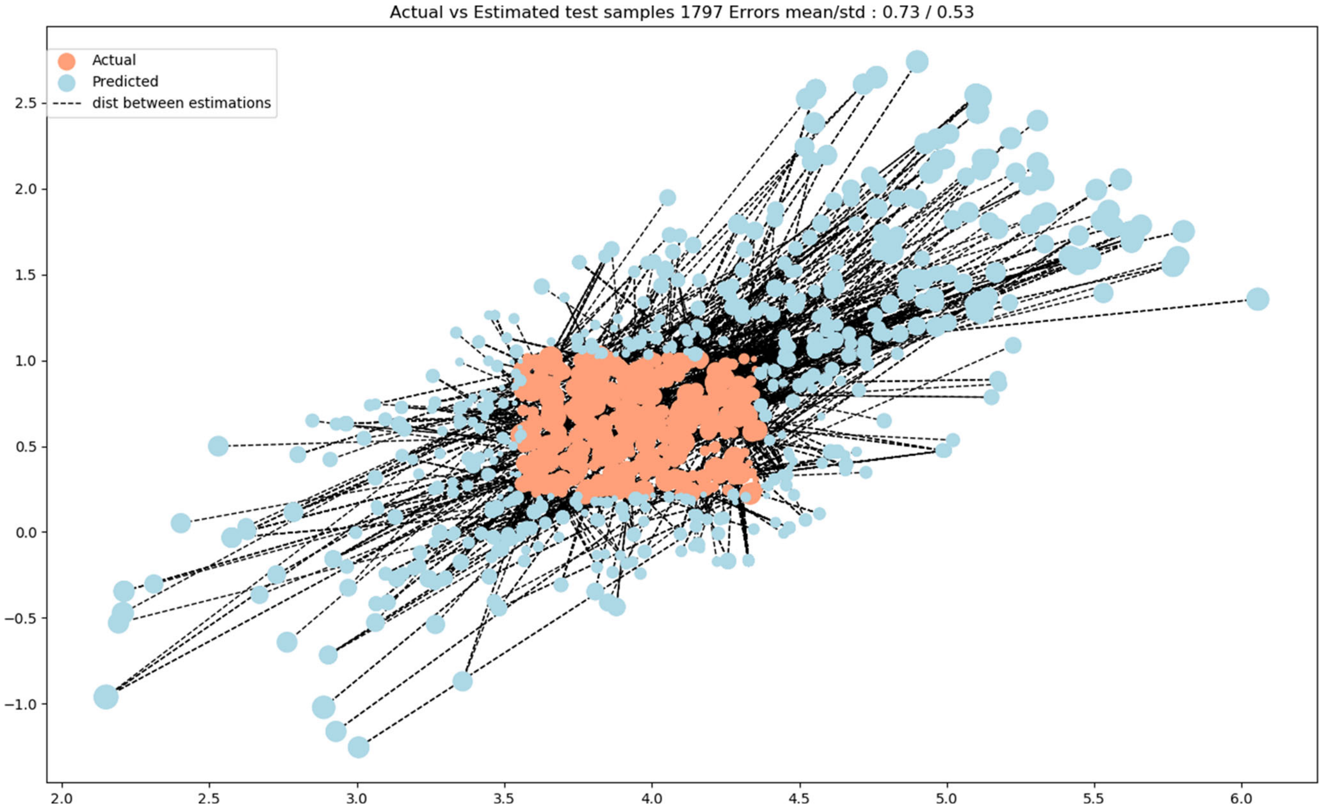


Fig. 5 KNN predicted positions vs actual positions for square test set

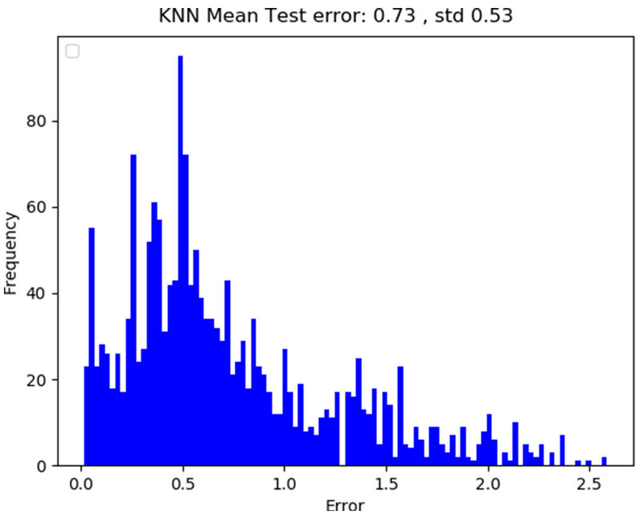


Fig. 6 KNN error distribution for square test

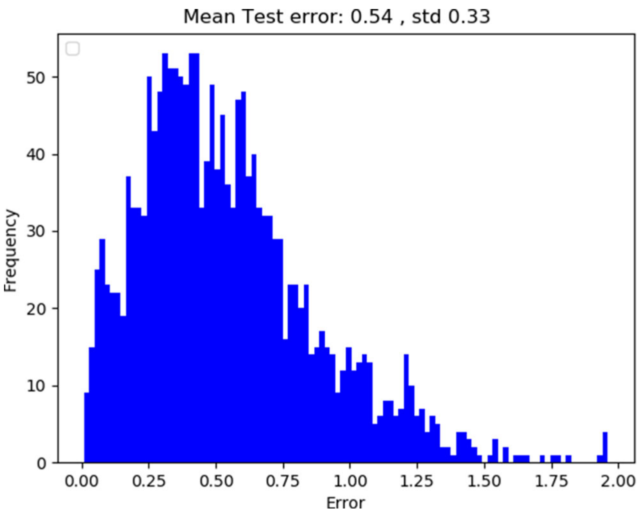


Fig. 8 MLP NN error distribution for square test set

0.54m and 0.33m, respectively, the error is still much larger than that of the random test sample selection. Even though the error distribution seems more structured than that of the KNN, it is difficult to draw a conclusion of a decent extrapolation ability of the model with such large error.

4.2.2 Sequential test set selection

The square test set selection is very difficult for the learning model to learn because the test set region is a relatively large blind spot. The model has to achieve a very challenging extrapolation outside the train set region to make adequate estimations within the square region.

In the sequential test set selection, we make use of the order in which the dataset is provided. As previously mentioned, the dataset was created by moving the transmitter along the table using a small vacuum cleaner robot. The dataset is provided in the order in which the transmissions were sent. This means that the path of the transmitter can be tracked by traversing the positions in order. We make use of

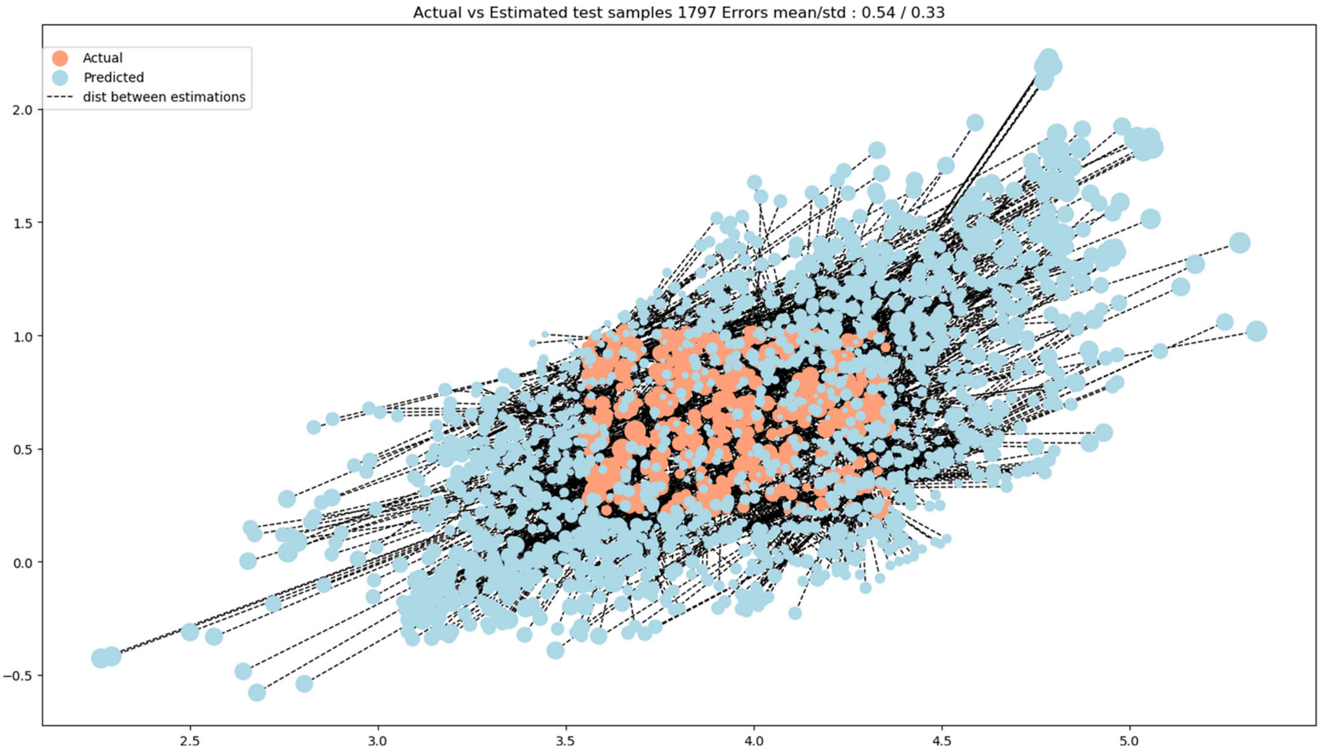


Fig. 7 MLP NN predicted positions vs actual positions for square test set

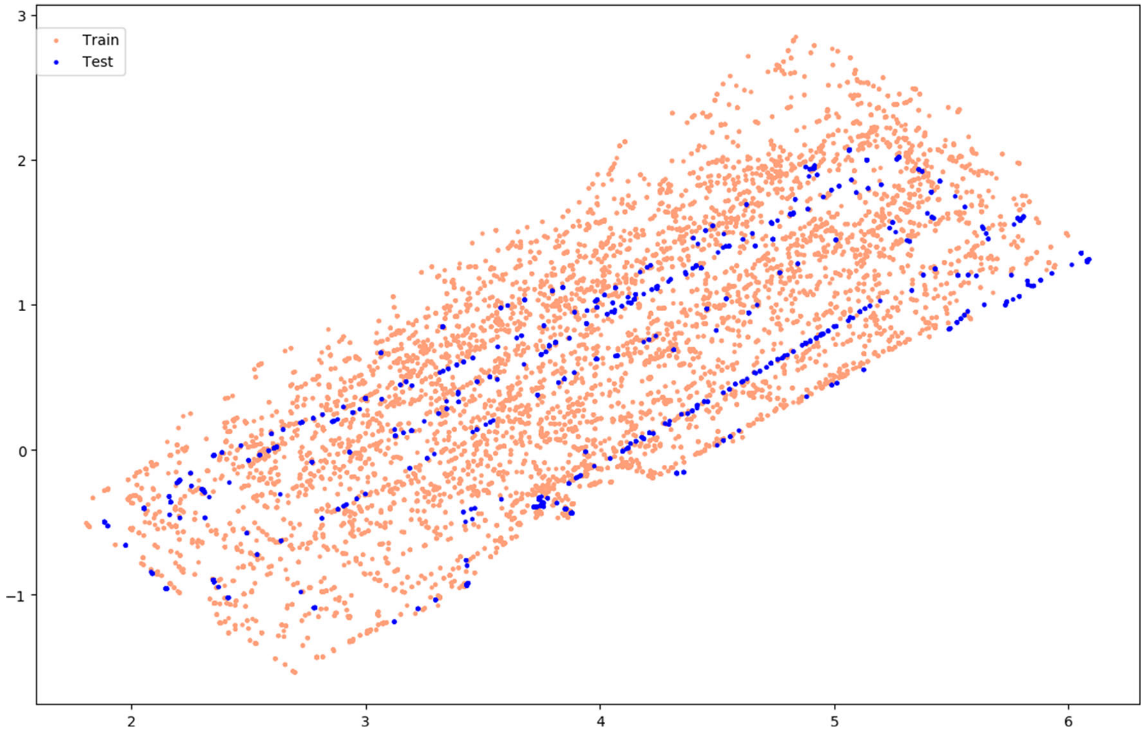


Fig. 9 Sequential test set selection

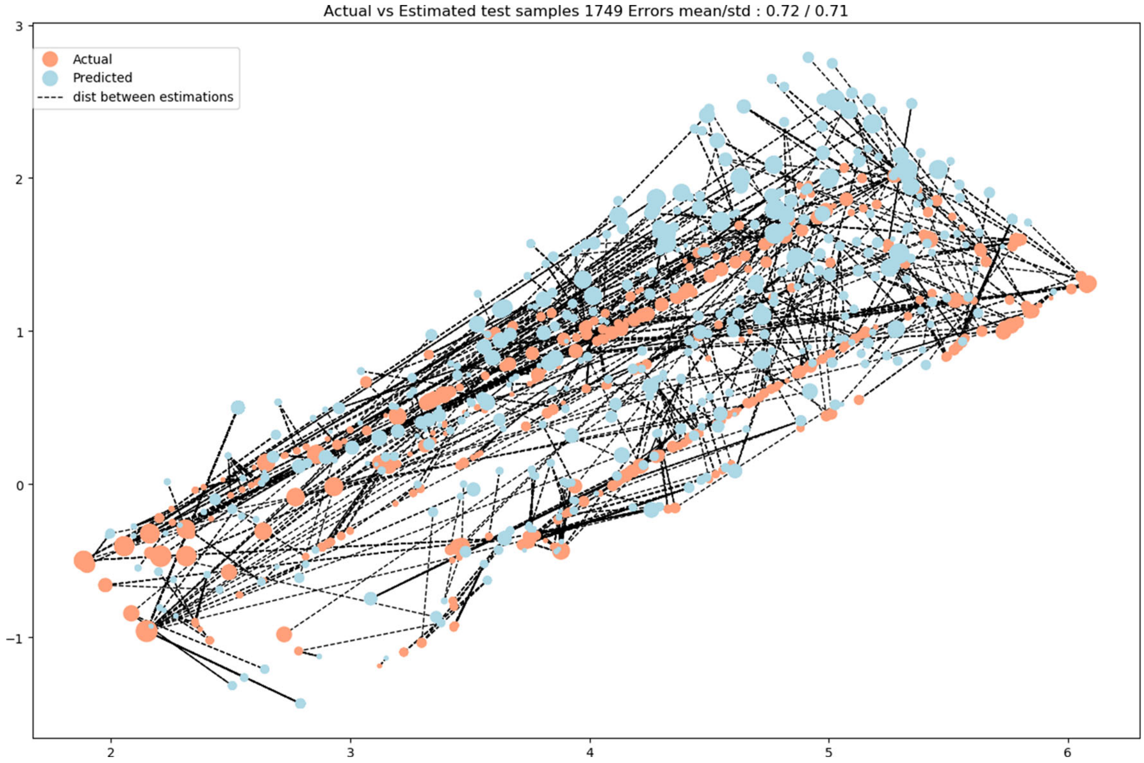


Fig. 10 KNN predicted positions vs actual positions for sequential test set

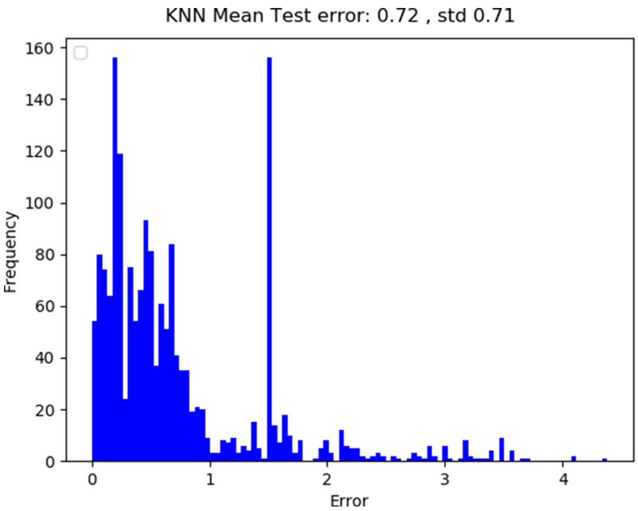


Fig. 11 KNN error distribution for sequential test

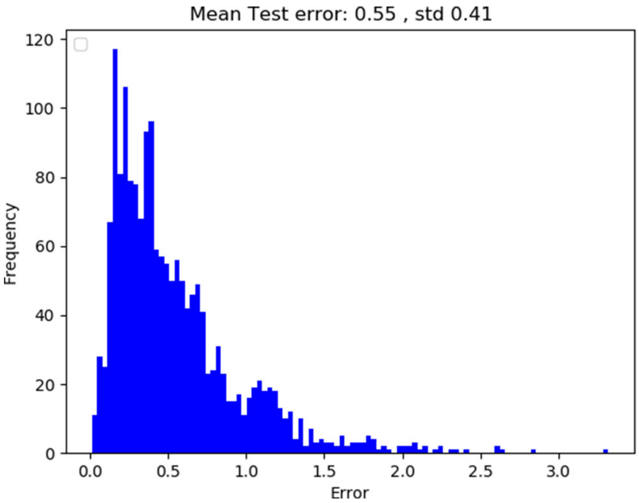


Fig. 13 MLP NN error distribution for sequential test

this order by setting the test set to be the first 10% of samples read sequentially in order from the provided dataset. This ensures that there is no intersection between the positions of the train and test sets since moving randomly along the table makes it improbable to visit the exact same position twice. Figure 9 demonstrates the train set and test set selection using the sequential method.

The test set samples in this selection method are relatively spread along the table. Test set positions are close to train set positions but not superposing. The prediction task appears easier than the square selection since the model does not need to extrapolate in a blind region. Rather, the model needs to be able to relate test set samples to their nearby training samples then interpolates to estimate the test

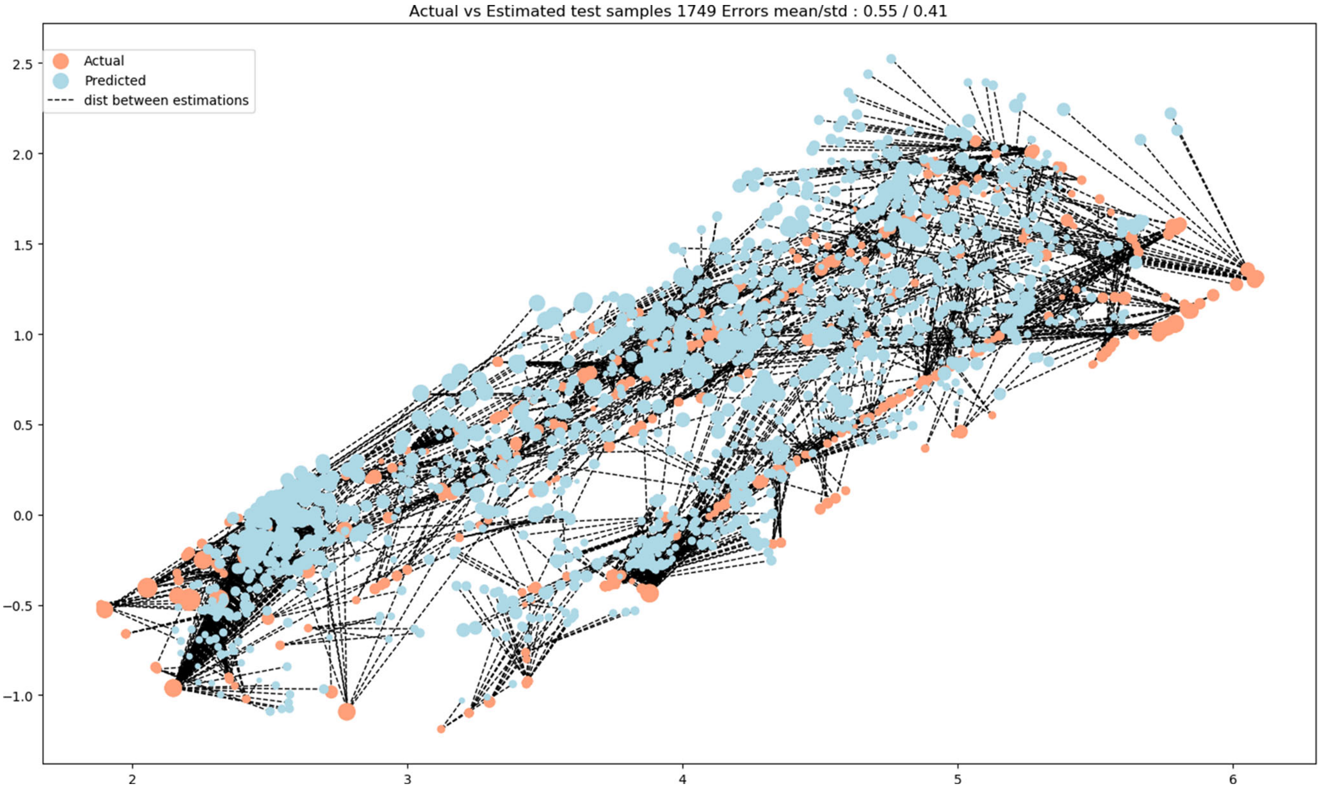


Fig. 12 MLP NN predicted positions vs actual positions for sequential test set

Table 2 Indoor vs outdoor characteristics

Aspect	Indoor [16]	Outdoor [17]
Measurements	CSI	CSI
Antennas	2 × 8	8 × 8
Subcarriers	924	924
Environment	Conference room	Residential area
Noise	Low	High
Area	Table: 8 m^2	Streets: several kms

sample position. We start with the KNN estimations, Fig. 10 shows the KNN's predictions and the corresponding ground truths of the test samples. KNN predictions are not far from the test set region since the training set samples are spread along the table. However, KNN still suffers from very large errors with an error of 0.72 m. The error distribution for the KNN estimations is shown in Fig. 11.

The error of the KNN model is as large as that of the square test selection. Using k values larger than 1 does not improve performance. The experiment is repeated using MLP NN; Fig. 12 shows the MLP NN predictions and the corresponding ground truths positions. The dispersed predictions prove that even the MLP NN is not able to perform well for the sequential test. This conclusion is backed up by the high mean square error of 0.55 m. The error distribution of the MLP NN method depicted in Fig. 13 appears more structured that the KNN error distribution. However, the error is as high as the experiment with the square test selection making it difficult to conclude that it has a better generalization ability.

4.2.3 Outdoor experiment

To further reinforce the obtained results, we tested the generalization aspect of machine learning models in an outdoor experiment prepared for the IEEE Communication Theory Workshop 2020 Data Competition [17]. The event was postponed due to the COVID-19 pandemic but the experimental dataset was shared with the scientific community. Both the indoor and outdoor experiments are CSI-based. Table 2 compares the characteristics of both experiments. The MIMO antenna used in this experiment is 8 × 8 which means more information is available for each transmission. However, the outdoor context introduces larger noise and NLoS transmissions. Figure 14 shows the map of the traversed region in the left subfigure and the Cartesian positions at which transmissions occurred in the right subfigure where the antenna is positioned at the origin.

The dataset is divided into 5k labelled samples and 36k unlabelled samples. In this paper, we focus on the supervised aspect using only labelled dataset. In order to cope with the richer information content, we propose the use of Fourier transform [19] to reduce noise and dimensionality instead of polynomial regression. This leads to a faster processing and higher control on the flexibility of the representing function. The Fourier series approximates any function in terms of infinite sums of sines and cosines

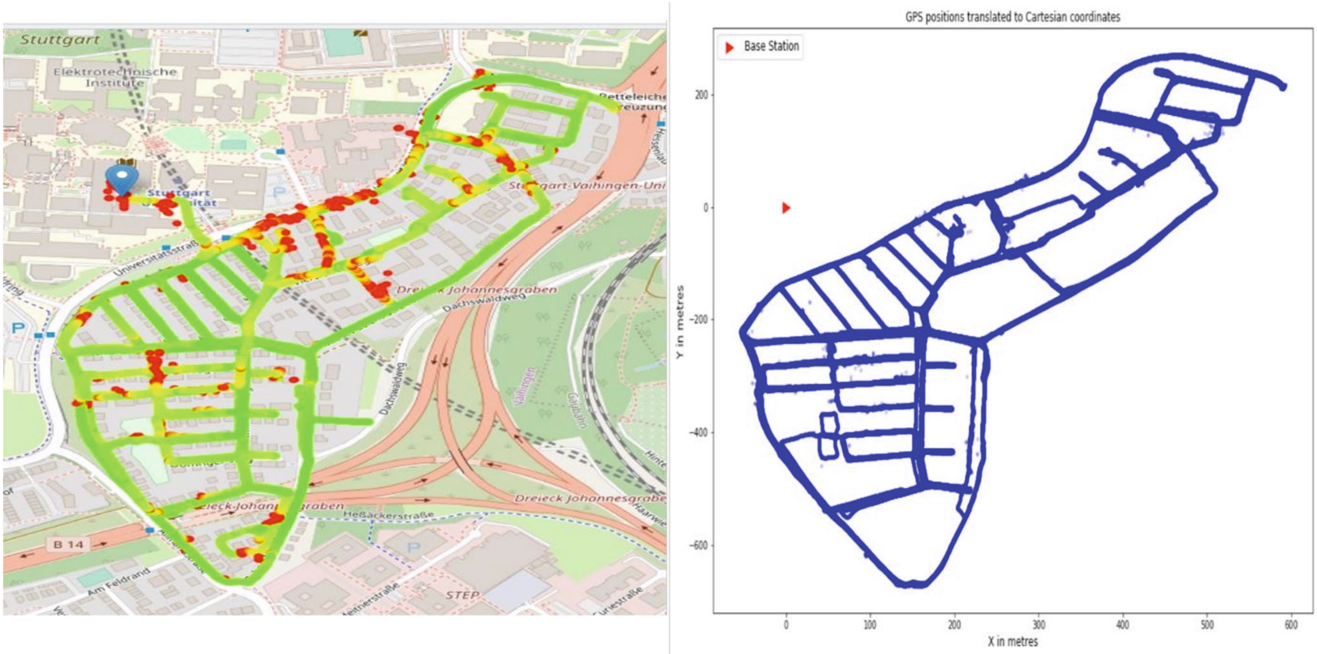


Fig. 14 Traversed roads in a residential area in Stuttgart [17]

Table 3 Performance of MLP NN and KNN with different preprocessing methods

Learning model	Preprocessing	RMSE
KNN	Polynomial regression	118 m
KNN	Fourier fitting	130 m
MLP NN	Polynomial regression	44 m
MLP NN	Fourier fitting	37 m

of increasing frequencies. Thus, we control the fluctuation flexibility by limiting the number of used sines and cosines. Furthermore, computation speed is improved with the use of the fast Fourier transform [20].

The labelled dataset is split into 90% training set and 10% test set while ensuring that both sets are not overlapping. Starting from the best solution in terms of error in the indoor positioning competition, we apply the K-nearest neighbor method to the labelled dataset. The value of k is set to one and the neighboring criterion is chosen to be the Euclidean distance between CSI magnitude component values. KNN is experimented with polynomial regression and Fourier fitting yielding a mean error of 118 m and 130 m, respectively. The high error can be due to the fact that the data is too sparse or that KNN is too primitive in nature to capture useful features. We believe that both factors contribute to the high error; this is evident by the performance of MLP NN on the labelled dataset. The MLP NN hyperparameters are the same as those of the highest performing MLP NN in the indoor experiment. The MLP NN achieves a mean error of 44 m with polynomial regression approximation of the CSI magnitude values and 37 m with the use of Fourier. The large gap between the error of KNN and MLP NN shows that MLP NN was able to capture more useful features that KNN's simplistic approach failed to detect. However, the fact that the error is still much higher than the accuracy of the differential GPS used to record the ground truth positions (< 1m) shows that the sparsity of the datapoints hinders the learning process significantly. Table 3 summarizes the experiments with KNN and MLP NN with some variation of the preprocessing steps.

5 Conclusion and future work

Prior to the generalization experiments, we did not expect the KNN to be able to generalize due to its simplistic approach in estimating test set samples. However, we had better expectations for the MLP NN's performance. The experimental results show relatively large errors for both models and it appears that both models fail to generalize. Nevertheless, there is a considerable difference between the mean errors of KNN and MLP NN. MLP NN's error is approximately 25% lower than that of KNN in the indoor experiment and 28% in the outdoor experiment. The consistent difference suggests that MLP NN has a higher potential for generalization over KNN.

The indoor experiment [16] helped to explain why KNN outperformed MLP NN in terms of absolute error when the test set was randomly selected. The sequential test selection experiment reveals that both KNN and MLP NN struggle to relate the test sample to nearby train samples. Yet, CSI samples measured from the same position are relatable by both models. Since the number of dataset samples is large with respect to the traversed area of the table, multiple transmissions occur at the same positions. When random test sample selection occurs, at least one of the repetitions is selected in the train set and at least one resides in the test set. Thus, at inference time, the learning model is able to relate one of the repetitions found in the learning phase to the test sample. However, the model fails when test set selection is applied in a way that separates the repetitions, making them all in either the train or test set. KNN focuses on relating the test set sample to the closest one in the train set which often turns out to be one of the repeated measurements at the same position. This gives the KNN method an edge over the MLP NN that, on the other hand, forms a highly complex non-linear function where similar CSI input will get similar but not superposing positions.

The outdoor experiment [17] required generalization to obtain low error values. With the focus only on the labelled dataset, the sparsity of measurements is a large obstacle for both learning models. MLP NN consistently performed better than KNN which supports the intuition obtained from the indoor experiment about the MLP NN's potential to generalize. In an attempt to further improve the localization accuracy, we attempted the use of CNN which showed an even higher potential for generalization that could be exploited in further research.

Possible future work to improve the generalization ability of the deep learning models is to bias the model towards learning relational features between nearby CSI readings. In image recognition context, CNNs induce this relational bias between nearby pixels depending on the size of the kernel. We believe that a more flexible model to induce such bias is Graph Neural Networks (GNN) which showed high generalization abilities in different contexts [8].

References

1. Čapkun S, Hamdi M, Hubaux JP (2002) GPS-free positioning in mobile ad hoc networks. Clust Comput 5(2):157–167
2. Sobehy A, Renault E, Muhlethaler P (2019) Position certainty propagation: a localization service for ad-hoc networks. Computers 8(1):6

3. Nandakumar R, Chintalapudi KK, Padmanabhan VN (2012) Centaur: locating devices in an office environment. In: Proceedings of the 18th annual international conference on mobile computing and networking. ACM, pp 281–292
4. Yang Z, Zhou Z, Liu Y (2013) From RSSI to CSI: Indoor localization via channel response. ACM Comput Surv (CSUR) 46(2): 25
5. He S, Chan S-HG (2015) Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons. IEEE Commun Surv Tutorials 18(1):466–490
6. Sobehy A, Renault É, Mühlethaler P (2019) CSI based indoor localization using ensemble neural networks. Machine Learning for Networking, Paris, France <hal-02334588>
7. Song Q, Songtao G, Xing L, Yuanyuan Y (2017) CSI amplitude fingerprinting-based NB-IoT indoor localization. IEEE Internet Things J 5(3):1494–1504
8. Battaglia PW, Hamrick JB, Bapst V, Sanchez-Gonzalez A, Zambaldi V, Malinowski M, Tacchetti A (2018) Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261
9. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
10. Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. arXiv:1409.0473
11. Arnold M, Hoydis J, Brink ST (2019) Novel massive MIMO channel sounding data applied to deep learning-based indoor positioning. In: SCC 2019; 12th international ITG conference on systems communications and coding. VDE, pp 1–6
12. Sobehy A, Renault E, Muhlethaler P (2019) NDR: Noise and dimensionality reduction of CSI for indoor positioning using deep learning. In: GlobeCom, Hawaii, United States
13. De Bast S, Pollin S (2020) MaMIMO CSI-based positioning using CNNs: Peeking inside the black box. ICC, Dublin, Ireland
14. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
15. Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vision 115(3):211–252
16. (2019) IEEE communication theory workshop 2019 indoor positioning competition. https://ctw2019.ieee-ctw.org. Accessed 24 Aug 2020
17. (2020) IEEE communication theory workshop 2020 IEEE CTW 2020 data competition. https://ctw2020.ieee-ctw.org. Accessed 10 Apr 2021
18. Wang X, Gao L, Mao S, Pandey S (2015) DeepFi: Deep learning for indoor fingerprinting using channel state information. In: 2015 IEEE wireless communications and networking conference (WCNC). IEEE, pp 1666–1671
19. Brunton S, Nathan Kutz J (2019) Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press, Cambridge
20. Nussbaumer HJ (1981) The fast Fourier transform. In: Fast Fourier Transform and Convolution Algorithms. Springer, pp 80–111