



Article Generalization Investigation for Artificial Intelligence-Based Positioning in IIoT

Qin Mu *, Muqing Wu and Huixuan Zhou

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China

* Correspondence: muqin@bupt.edu.cn

Abstract: In recent years, IIoT scenarios show increasing demand for high-precision positioning. Cellular-based positioning plays a very important role in this application area. The cellular-based positioning system can be categorized as an asymmetric processing system. In this system, multiple base stations transmit positioning reference signaling to one device and the device derives its coordinates based on the measurement of the positioning reference signal by using certain positioning algorithms. The classical positioning algorithms encounter great challenges in satisfying the stringent precision requirement. Meanwhile, artificial intelligence (AI)-based positioning solutions draw great attention due to their strong ability in improving positioning accuracy in IIoT scenarios. For AI-based algorithms, generalization capability is one important metric. It demonstrates the ability to adapt to different environments. However, there is little literature touching on the investigation of generalization capability. In this article, we will tackle this issue by considering typical features in IIoT scenarios including the clutter distribution, network synchronization error, and receiving timing error. The impact of these typical features on the generalization capability is firstly evaluated and then the feasibility of existing popular generalization improvement solutions, i.e., optimized training data set and fine-tuning are tested under different cases. At last, directions to further guarantee the generalization capability are presented. The results of this article provide useful experience for developing AI models for positioning in realistic IIoT scenarios.

Keywords: positioning; AI; clutter distribution; network synchronization; receiving timing; generalization

1. Introduction

In recent years, positioning plays a more and more important role in both the consumer market and the vertical industries. In the consumer market, the positioning enables the services such as shopping guides in shopping malls, and self-service tour guides. In vertical industries, positioning could facilitate intelligent manufacturing and emergency rescues [1]. In all the positioning services especially in the vertical industries, high positioning precision is one crucial requirement.

In the 5G defined in the 3rd Generation Partnership Project (3GPP), positioning technology is always an important technique field. In the 5G system, 3GPP specified multiple positioning solutions such as timing difference of arrival (TDOA), and multi-cell round trip time (Multi-RTT) [2] to enable the positioning service. The target scenarios in 5G release 16 are mainly the commercial scenarios and the target accuracy requirements are defined as follows [3]:

- Horizontal positioning error < 3 m for 80% of devices in indoor deployment scenarios;
- Vertical positioning error < 3 m for 80% of devices in indoor deployment scenarios;
- Horizontal positioning error < 10 m for 80% of devices in outdoor deployments scenarios;
 - Vertical positioning error < 3 m for 80% of devices in outdoor deployment scenarios.



Citation: Mu, Q.; Wu, M.; Zhou, H. Generalization Investigation for Artificial Intelligence-Based Positioning in IIoT. *Symmetry* **2023**, *15*, 992. https://doi.org/10.3390/ sym15050992

Academic Editors: Peng Li, Chao Fang and José Carlos R. Alcantud

Received: 18 March 2023 Revised: 8 April 2023 Accepted: 25 April 2023 Published: 27 April 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). In 5G release 17, enhancement of the positioning solutions was carried out. The main target scenarios are the IIoT scenarios and the defined target accuracy requirements are as follows [4]:

- Horizontal position accuracy (<0.2 m) for 90% of devices
- Vertical position accuracy (<1 m) for 90% of devices

However, according to the evaluation and study in release 17, it is observed that it is quite challenging to achieve the high precision required in IIoT scenarios by using traditional positioning algorithms. The main reason is that the achieved accuracy of traditional positioning algorithms highly depends on the number of line of sight (LOS) paths. While, in the IIoT scenario, due to the distribution of different blocks, the probability of the LOS path is quite low. In addition, timing-based positioning solutions such as TDOA or multi-RTT are quite sensitive to timing synchronization. However, the network synchronization error and the receiving timing error on the device side would further degrade the positioning accuracy.

These years, artificial intelligence (AI) technology has developed rapidly. It shows the excellent capability to handle complicated processing. Researchers consider utilizing this tool to improve positioning accuracy [5,6]. The efficient feature extraction function of AI can be used to extract the abstract features of the measurement information of the channel from the base station to the user, which may bring a major breakthrough to solve the above problems.

For AI-based solutions, generalization capability is one important metric. It demonstrates the ability to adapt to different environments. Good generalization capability guarantees performance in various environments. In the IIoT scenarios, the clutter distribution, network synchronization, and receiving timing error on the device side may vary in time or with the change in indoor layout. However, currently, there is no literature to investigate the generalization capability against the change in clutter distribution, network synchronization, and receiving timing errors. In this paper, we will fill this blank area.

This article will first evaluate the generalization performance of AI-based positioning when changing the clutter distribution, network synchronization error and device receiving timing error, respectively. Then, existing popular solutions to improve the generalization will be studied and compared to figure out the feasibility in IIoT scenarios. At last insightful observations would be drawn, which would be good guidance for the model development in realistic scenarios.

The structure of the paper is as follows. Related works are summarized in Section 2. The system model is introduced in Section 3. Section 4 explains the evaluation setting and methods to be investigated. In Section 5, we test and analyze the impact on generalization capability in different scenarios. Section 6 concludes this paper.

2. Related Works

Utilizing AI approaches to improve positioning accuracy is boosting these years. Some of them use AI-based schemes such as neural networks with a single hidden layer [7–9], deep belief networks [10,11], deep neural networks [12,13], Gaussian regression [14], fuzzy neural networks [15], artificial synaptic networks [16], support vector machines (SVM) [17], random decision forest [18], or combinations of them [19]. Iqbal et al. [20] correlate measurement in a clinical environment by using a convolutional neural network (CNN). Ibrahim et al. [21] use CNNs and Sahar et al. [22] use long short-term memory (LSTM) recurrent neural networks (RNN) to process a time series of radio signaling measurements. Feigl et al. [23] use LSTMs on ToAs in a TDoA system setup. Some recent work even estimates positions in reinforcement learning settings [24]. These works mainly target to improve positioning accuracy or improve processing efficiency.

However, data-driven localization easily suffers from environmental changes. To make the AI model more robust against environmental change, there are works on utilizing transfer learning or optimized data set construction. Firstly, optimizing the data set construction is one well-known approach. In this approach, the data set is collected from more diverse scenarios [25]. Secondly, another approach is using transfer learning, in which an AI model is pre-trained and then the model is tuned by the newly collected data in the target scenario. In [26], the CNN model is pre-trained and then adapts the trained model to a new configuration in LOS conditions by using simulated data rather than realistic field data to reduce the site survey overhead. In addition, [27] considers a similar manner for the adaptation to a new environment. The difference is that it considers the NLOS condition. To reduce the effort of data collection in a realistic network, [28,29] consider a solution in another way around. In these works, the AI model is firstly pre-trained by using a simulated data set and then fine-tuned by using a small amount of data from a realistic network. However, all these works just consider how to apply transfer learning in an effective way and focused scenarios are not IIoT scenarios that encounter significant gaps in the accuracy requirement. In addition, these works lack analysis of individual factors affecting the generalization performance. To the best of our knowledge, our paper is the first one to investigate the impact on generalization capability from individual typical factors in IIoT scenarios.

Compared with existing works, the contribution of our paper is summarized as follows:

- Identify typical factors affecting the generalization performance and then evaluate the exact impact;
- Verify the feasibility of existing generalization solutions to reduce the side effect of variation of the typical factors;
- For the cases where existing generalization solutions do not help much, we further
 propose directions to improve the performance.

3. System Model

As explained in Section 1, the most challenging scenario is the IIoT scenario where the probability of LOS is extremely low and classical positioning methods hardly satisfy the precision requirement. In this paper, we will focus on this scenario.

3.1. Scenario

Usually, the IIoT scenario mainly locates indoors. In the indoor space, the clutter is distributed. The clutter could be big machinery composed of regular metallic surfaces. For example, several mixed production areas with open spaces and storage/commissioning areas or small to medium metallic machinery and objects with irregular structures. For example, assembly and production lines are surrounded by mixed small-sized machinery.

As for the base station (BS) layout, we refer to the typical deployment in 3GPP as shown in Figure 1. In this case, 18 BSs are distributed within a square lattice with spacing D, located D/2 from the walls. Here, we consider the small hall with L = 120 m, W = 60 m, and D = 20 m. Positioning devices are uniformly distributed over the convex hull of the horizontal BS deployment.



Figure 1. Typical BS layout.

The positioning procedure is a certain asymmetric procedure as shown in Figure 2. Multiple BS would transmit positioning reference signals (PRS) to the devices. On the device side, the positioning device would derive its coordinates according to the measurement of PRS from multiple BSs based on positioning algorithms.



Figure 2. Positioning procedure.

3.2. Modeling the LOS/NLOS Probability Based on Clutter Parameter

Due to the distribution of clutters in the factory, radio propagation would be impacted. In this paper, the clutter parameters include the density of the clutter, the height of the clutter, and the size of the clutter. High clutters would result in the potential NLOS propagation and dense clutters would further cause serious multipath effects as shown in the example of Figure 3.



Figure 3. Impact on radio propagation from clutter distribution.

In this paper, we refer to the modeling of the LOS probability in TS 38.901 by considering the clutter parameters [30]. Additionally, the LOS probability is modeled as

$$\Pr_{LOS,subsce}(d_{2D}) = \exp\left(-\frac{d_{2D}}{k_{subsce}}\right)$$
(1)

where

$$k_{subsce} = \frac{d_{clutter}}{\ln(1-r)} \cdot \frac{h_{BS} - h_{UT}}{h_c - h_{UT}}$$
(2)

Accordingly, for one specific link, the NLOS probability is $1 - Pr_{LOS,subsce}(d_{2D})$.

In the equation, r, $d_{clutter}$, h_c represent the clutter density, size of clutter, and height of clutter, respectively. In addition, h_{BS} and h_{UT} denote the height of BS and the height of the device, respectively.

3.3. Modeling of Network Synchronization Error

In the practical deployment, it is hard to ensure ideal synchronization among all the BSs used for positioning. In this study, we model the network synchronization error according to TR 38.857. The network synchronization error, per UE dropping, is defined as a truncated Gaussian distribution of zero mean and standard deviation of T1 ns values between a BS and a timing reference source which is assumed to have perfect timing, subject to the largest timing difference of T2 ns, where $T2 = 2 \times T1$

3.4. Modeling of Receiving Timing Error

The result of receiving time delay involved in the reception of a signal before reporting measurements are obtained from the signal. It is the uncalibrated receiving time delay, or the remaining delay after the device's internal calibration/compensation of the receiving time delay, involved in the reception of the positioning signals. The calibration/compensation may also include the calibration/compensation of the relative time delay between different RF chains in the same device and may also possibly consider the offset of the receiving antenna phase center to the physical antenna center.

The timing errors are modeled according to TR 38.857, which means that the device receiving timing error is modeled as one truncated Gaussian distribution with zero mean and standard deviation of T1 ns, with a truncated range as [-2T1,2T1]. As a clarification, the device receiving timing error is generated randomly for each device drop but is fixed for all BSs linked with this device, while the network synchronization errors are independent and identically distributed among 18 BSs and generated randomly for each device dropping.

4. Evaluation Set Up

In this section, we will introduce the evaluation methodology and the methods to be investigated.

4.1. AI-Based Positioning Methods

In this paper, two kinds of AI-based positioning methods are considered for evaluation. The first one is the direct AI-based positioning as shown in Figure 4, in which the output of the AI model is device's coordinates, and the other one is the indirect AI-based positioning, in which the output of the AI model is the intermediate parameter for the positioning. For example, the output of the AI model could be the ToA, Angle of Arrival (AoA), or the LOS or NLOS identity. Then, derive the UE's coordinates based on the inference intermediate parameter by the traditional positioning. In this paper, we will use the AI model to predict ToA. Then, using the TDOA algorithm to calculate the coordinates as shown in the principle in Figure 5. In both AI-based positioning methods, the input of the AI model is the channel impulse response (CIR) of the received positioning reference signaling from 18 BSs.



Figure 4. Direct AI/ML-based positioning.



Figure 5. Indirect AI/ML-based positioning: AI/ML-based ToA prediction.

In the practical system, one possible deployment solution is an AI model trained and then deployed on the device side. The BSs transmit positioning reference signal (PRS) to the devices and the devices perform the measurements. For direct AI-based positioning, devices could input the obtained measurement result into the AI model to infer the device's coordinates as shown in Figure 4. For the indirect AI-based positioning, UE could first obtain the ToA by using the AI model and then obtain the final coordinates by using the inference ToA and TDOA as shown in Figure 5.

4.2. Data Set Generation

The data sets include the data sets for training and the data sets for tests. In our paper, all data sets are generated by the simulation platform based on the methodology

defined in [4] and the parameter in Table 1. To investigate the impact of different practical parameters including the clutter parameter, network synchronization parameter, and the receiving error parameter, multiple data sets with parameter variation would be constructed. In all data sets, each sample contains the input and the label. In all the data sets, the input is the channel impulse response of PRS from 18 BSs. For direct AI-based positioning, the label is the device's coordinates and for indirect AI-based positioning, the label is the 18 ToAs for 18 BSs, respectively.

The detailed parameters for the data set are summarized as follows:

Parameters		Value
Carrier freq	uency, GHz	3.5 GHz
Bandwid	lth, MHz	100 MHz
Subcarrier s	pacing, kHz	30 kHz for 100 MHz
Channe	l model	InF-DH
	Hall size	$120 \times 60 \text{ m}$
Layout	Room height	10 m
	Number of floors	1
Total gNB TX	Cpower, dBm	24 dBm
gNB antenna configuration		$(M, N, P, Mg, Ng) = (4, 4, 2, 1, 1), dH = dV = 0.5 \lambda$
gNB antenna radiation pattern		Single sector
gNB antenna height		8 m
Penetration loss		0 dB
UE antenna configuration		$Mg = 1, Ng = 1, P = 2, dH = 0.5 \lambda, (M, N, P, Mg, Ng) = (1, 2, 2, 1, 1)$
UE horizontal drop procedure		Uniformly distributed over the convex hull of the horizontal BS deployment.
UE antenna height		1.5 m
UE mobility		3 km/h
Min gNB-UE distance (2D), m		0 m
UE antenna radiation pattern		Omni, 0 dBi
Clutter parameters: {density r , height h_c , size $d_{cluster}$ }		High clutter density: {40%, 2 m, 2 m} {60%, 6 m, 2 m}

Table 1. Parameter setting for the data set generation.

4.3. AI Model Construction and Model Training

In pre-tests, we evaluated several different well-known deep learning architectures such as GoogLeNet, DenseNet, and ResNet. As shown in Figure 6, it turned out that the ResNet provides higher positioning accuracy. ResNet is a feedforward neural network that employs residual mapping and has great performance on image feature extraction. In ResNet, the processing is not purely layer by layer. The processing of certain layers would be skipped to weaken the strong correlation between each layer, which is called a shortcut connection. Benefitting from its residual framework, ResNet solves the vanishing gradient and explosion problems that occur with the superposition of convolutional layers and pooling layers in deep network structures. Our algorithm refers to the 34-layer ResNet structure in [31], basically composed of 3, 4, 6, and 3 residual modules (building blocks as

shown in different colors in Figure 7) with convolutions of 64, 128, 256, and 512 channels, respectively. We made the following changes to the ResNet architecture in order to facilitate our task:

- In module conv1, we modify the convolutional layer to have a kernel size of 3 instead of 7 and a stride of 1 instead of 2. This is because the dimension of channel features of positioning data is 18 × 256 rather than 224 × 224 in [31] as input in the original model, so the size of the convolution kernel needs to be reduced. We set the stride to 1, and do not carry out downsampling in the first two layers so that the dimensions of the original input can be preserved. Additionally, the subsequent layers perform downsampling like the original model, which is conducive to extracting 18-dimensional BS information;
- In module conv2_x, we remove the max-pooling operation, and we set the convolutional layers to have a stride of 1 instead of 2.



Figure 6. Positioning accuracy comparison among different types of AI models.



Figure 7. AI model structure.

After module conv5_x, we replace the fully connected (FC) layer that has 1000 output units (before the classifier) with an FC layer of 2 units outputting a vector of positions (x, y) or 18 units for a vector of device ToAs. We further remove the SoftMax classifier. The detailed AI model structure is shown in Figure 7.

The positioning problem to be solved is regression, not classification, so the loss function is set to mean squared error (MSE). For training, we deploy an adaptive moment estimation (Adam) optimizer. As for the learning rate, we test several candidate values including 0.005, 0.001, 0.0005, and 0.0001. We found learning rate of 0.001 could achieve a better balance between performance and converging rate. So, in the following evaluation,

the learning rate is set as 0.001. As for the batch size, we also test the batch size of 16, 32, 64, and 128. All these batch sizes achieve similar performance. So, we choose the middle value of 32 for the following evaluation. Regarding the epochs, we observed that when the number of epochs is around 200, the positioning accuracy becomes relatively stable. Thus, in our evaluation, each model is trained by 200 epochs.

The model training is deployed on a computer with NVIDIA GeForce RTX 3090 GPU and Intel(R) Xeon(R) Platinum 8255C CPU@2.5 GHz. We have implemented all our DL framework in python3.8 on Ubuntu20.04 and use the pyTorch1.10 deep learning framework.

4.4. Training Manners to Be Evaluated

To evaluate the generalization performance, three training manners are to be evaluated.

- Manner 1: AI model trained by data samples with identical configurations. It is expected that the generalization of this manner is not good when the testing data set is generated based on different configurations from that of the training data set.
- Manner 2: The second one is the AI model is trained by data with mixed samples. The samples in the training data set are generated by different parameters, e.g., by different clutter parameters or by different timing error assumptions. The core idea of this manner is to optimize the data set construction to guarantee generalization performance.
- Manner 3: The third one is that the AI model is initially trained by data set with identical configuration and then fine-tuned by a small number of samples with the same configuration as that in the test data set. This manner has a similar spirit of transfer learning.

5. Performance Evaluation and Discussion

In this section, a simulation is carried out to investigate the performance of training manners explained in Section 4 by considering both direct positioning and indirect positioning. To observe the impact of different parameters on the generalization in the IIoT scenario, simulations are performed in different scenarios including the clutter parameters, network synchronization error, and receiving timing error. In addition, performance in identical configuration between the training data set and test data set is also evaluated for reference.

In the evolution, the metric is defined as the positioning error which is the distance between the coordinates derived based on the AI model and the ground truth. In addition, the positioning at the 90% percentile is also observed to check whether the precision requirement can be satisfied or not.

5.1. Scenario 0: Identical Configuration between Training Data Set and Test Data Set

In this scenario, the training data set and test set are generated in the same configuration, including the same clutter parameter setting, ideal network synchronization and no receiving timing error. Results in this scenario can be regarded as the upper bound.

Figures 8 and 9 show the positioning error for direct AI-based positioning and AIbased ToA prediction, respectively. For comparison, the positioning error of the classic positioning method TDOA is also depicted for reference.

According to the simulation results, In the TDOA-based method, the positioning error at 90% CDF for the clutter parameter of {60%, 6 m, 2 m} is up to 14 m, and even in the scenario of clutter parameter of {40%, 2 m, 2 m}, the positioning error at 90% CDF is still up to 12 m. While for the AI-based solutions, the performance is improved greatly. Depending on the direct AI-based solution or indirect AI-based solution used, the performance is slightly different. Anyway, no matter which AI-based solution is used, the positioning error @90% is less than 1 m. In the clutter parameter of {60%, 6 m, 2 m}, the positioning error could achieve around 0.45~0.65 m if using the AI model trained by the data set with clutter parameter of {60%, 6 m, 2 m}. In the clutter parameter of {40%, 2 m, 2 m}, the positioning error could achieve around 0.75~0.85 m if using the AI model trained by the data set of inF-DH{40%, 2 m, 2 m}.



Figure 8. Performance of direct AI/ML-based positioning with the same clutter parameters between training and test.



Figure 9. Performance of AI/ML-based ToA prediction with the same clutter parameter between training and test.

5.2. Scenario 1: Impact of the Clutter Parameter

Training manner 1: Different clutter parameters for training and test

We considered the clutter parameter of {60%, 6 m, 2 m} and the clutter parameter of {40%, 2 m, 2 m}. Two training data sets are generated. One is the data set purely generated based on the clutter parameter of {60%, 6 m, 2 m}, and another one is the data set purely generated based on the clutter parameter of {40%, 2 m, 2 m}. Two AI models are trained based on these two data sets, respectively. For the test, data sets of different clutter parameters are used.

Figures 10 and 11 show the evaluation results for the direct AI-based positioning and the AI-based ToA prediction. If the same clutter parameter is set for the training data set and the test data set, optimal positioning accuracy could be achieved for both direct AI-based positioning and AI-based ToA prediction as shown in scenario 0. While, once the clutter parameter for the test data set is different from that of the training data set, the inference performance degrades sharply. In the direct AI-based positioning, take the AI model trained by clutter parameter of {60%, 6 m, 2 m} data set as an example. The positioning error @90% is less than 0.5 m when using clutter parameter {60%, 6 m, 2 m} test data set, while the positioning error @90% is up to 7 m when using clutter parameter {40%,

2 m, 2 m} test data set. That is to say, a lack of generalization capability would happen if the data set was only generated based on limited clutter parameters.



Figure 10. Performance of direct AI/ML-based positioning with different clutter parameters between training and test.



Figure 11. Performance of AI/ML-based ToA prediction with different clutter parameters between training and test.

Training manner 2: Mixed clutter parameter setting for training

In this case, the data set is generated by mixing the samples from the clutter parameter of $\{60\%, 6 \text{ m}, 2 \text{ m}\}$ and clutter parameter of $\{40\%, 2 \text{ m}, 2 \text{ m}\}$. Additionally, the AI model is trained by the mixed data set and tested by the data set with clutter parameter of $\{40\%, 2 \text{ m}, 2 \text{ m}\}$ and test data set from clutter parameter of $\{60\%, 6 \text{ m}, 2 \text{ m}\}$, respectively.

Figures 12 and 13 summarize the evaluation results. Compared with the results in Figures 10 and 11, the positioning accuracy is improved greatly. On the other hand, compared with the performance of using the same clutter parameter, there is still a gap. However, from the perspective of whether satisfy the precision requirement, the positioning accuracy at 90% CDF is less than 1 m for both direct AI/ML-based positioning and AI/ML-based ToA prediction. That is to say, for the AI model trained with mix data set, the generalization problem can be relieved.



Figure 12. Performance of direct AI/ML-based positioning with mixed clutter parameters for training.



Figure 13. Performance of AI/ML-based ToA prediction with mixed clutter parameters for training.

Training manner 3: Fine-tuning

In this case, the model is firstly trained by data set with a clutter parameter of {60%, 6 m, 2 m} or a clutter parameter of {40%, 2 m, 2 m} and then fine-tuned by a data set with the same clutter meter with that of the test data set. The data set for fine-tuning only contains a small number of samples.

Figures 14 and 15 summarize the results for fine-tuning. Compared with results without fine-tuning in Figures 10 and 11, the positioning accuracy is improved reasonably. For example, for the AI model trained by clutter parameter {60%, 6 m, 2 m} and fine-tuned by data set with clutter parameter {40%, 2 m, 2 m}, when applying this model in the scenario with clutter parameter of {40%, 2 m, 2 m}, the positioning error is reduced from to around 7 m to around 1.5 m in both direct AI-based positioning method and AI-based ToA prediction method. However, compared with the results of the AI model trained with mixed data set in Figures 12 and 13, there is still some gap. The possible reason is fewer samples from the applied scenario for fine-tuning.



Figure 14. Performance of direct AI/ML-based positioning with fine-tuning.



Figure 15. Performance of AI/ML-based ToA prediction with fine-tuning.

5.3. Scenario 2: Impact of Non-Ideal Network Synchronization

In this section, we test the generalization performance with non-ideal network synchronization.

• Training manner 1: Training data set with ideal network synchronization and test data set with network synchronization error

Firstly, we test the case in which the AI/ML model is trained by a data set with ideal network synchronization and then is tested by data set with network synchronization error with a standard deviation of 50 ns. The results are summarized in Figures 16 and 17. It is observed that no matter in the direct AI-based positioning method or the AI-based ToA prediction method or no matter with clutter parameter {60%, 6 m, 2 m} or with clutter parameter {40%, 2 m, 2 m}, the positioning performance is quite poor and the positioning error at 90% CDF is more than 10 m.



Figure 16. Performance of direct AI/ML-based positioning with different network synchronization settings between training and test.



Figure 17. Performance of AI/ML-based ToA prediction with different network synchronization settings between training and test.

Training manner 2: Mixed network synchronization error setting for training

In this case, the data set for training is constructed by mixing data samples with different synchronization error settings. In the evaluation, standard deviations of 50 ns, 40 ns, 30 ns, and 0 ns are used for the training data set generation. Figures 18 and 19 show the performance for direct AI-based positioning and AI-based ToA prediction when the test data set is generated by using a 50 ns standard deviation. According to the simulation results, it is observed that, compared with training manner 1, using mixed network synchronization error setting could improve the positioning performance. Yet, on the other hand, the improvement is not so significant. For clutter parameter {40%, 2 m, 2 m}, the positioning error @90% is around 3 m for both direct AI-based positioning and AI-based ToA prediction, which is a little far from the requirement in the IIoT scenario.



Figure 18. Performance of direct AI-based positioning with mixed network synchronization error for training.



Figure 19. Performance of AI-based ToA prediction with mixed network synchronization for training.

Training manner 3: Fine-tuning

For each clutter parameter setting, the corresponding AI model is first trained by using ideal network synchronization. Then, fine-tune the AI model by a using small number of samples generated with the same network synchronization setting as that of the test data set. Figures 20 and 21 show the performance for direct AI-based positioning and AI-based ToA prediction when the AI model is fine-tuned by using data set with a 50 ns standard deviation and using test data set with a 50 ns standard. It is observed that using fine-tuning could improve positioning performance to some extent. Yet the improvement is quite limited. In addition, the positioning error at 90% CDF is around 5 m and 7 m for clutter parameters of {60%, 6 m, 2 m} and {40%, 4 m, 2 m}, respectively. There is still a large gap between the requirement of IIoT. Moreover, compared with the training manner 2, the improvement is small due to fewer training samples with similar features to that of the test data set.



Figure 20. Performance of direct AI-based positioning with samples with network synchronization error for fine-tuning.



Figure 21. Performance of direct AI-based ToA prediction with samples with network synchronization error for fine-tuning.

5.4. Scenario 3: Non-Ideal Receiving Timing

 Training manner 1: Training data set with ideal receiving timing and test data set with Receiving timing error

Firstly, we test the case in which the AI/ML model is trained by a data set with ideal receiving timing and then is tested by a data set with a 10 ns receiving timing error. The results are summarized in Figures 22 and 23. It is observed that no matter in the direct AI-based positioning method or the AI-based ToA prediction method or no matter with clutter parameter {60%, 6 m, 2 m} or with clutter parameter {40%, 2 m, 2 m}, the positioning performance degrades. Compared with other scenarios, the degradation is relatively small.

Training manner 2: Mixed receiving timing error setting for training

In this case, the data set for training is constructed by mixing data samples with different receiving timing error settings. In the evaluation, standard deviations of 10 ns, 20 ns, 30 ns, and 0 ns are used for the training data set generation. Figures 24 and 25 show the performance for direct AI-based positioning and AI-based ToA prediction when the test data set is generated by using 10 ns standard deviations. According to the simulation

results, it is observed that, compared with training manner 1, using mixed receiving timing error setting could improve the positioning performance greatly.

Training Manner 3: Fine-Tuning

For each clutter meter setting, the corresponding AI model is first trained by using ideal receiving timing. Then, fine-tune the AI model by using a small number of samples generated with the same receiving timing error setting as that of the test data set. Figures 26 and 27 show the performance for direct AI-based positioning and AI-based ToA prediction when the AI model is fine-tuned by using data set with a 10 ns standard deviation and using test data set with a 10 ns standard deviation. It is observed that using fine-tuning could improve positioning performance to some extent. Yet the improvement is not significant. In addition, the positioning error at 90% CDF is around 1.7 m and 2 m for clutter parameters of {60%, 6 m, 2 m} and {40%, 2 m, 2 m}, respectively. There is still some gap in the requirement of IIoT. Moreover, compared with the training manner 2, the improvement is small.



Figure 22. Performance of direct AI-based positioning with different receiving timing error settings between training and inference.



Figure 23. Performance of AI-based ToA prediction with different receiving timing error settings between training and inference.



Figure 24. Performance of direct AI-based positioning with mixed receiving time errors for training.







Figure 26. Performance of direct AI-based positioning with samples with receiving timing error for fine-tuning.



Figure 27. Performance of AI-based ToA prediction with samples with receiving timing error for fine-tuning.

5.5. Discussion and Summary

According to the simulation results above, the following summary can be made:

- In the IIoT scenario, the AI-based positioning methods are sensitive to the change in clutter parameters, network synchronization errors, and receiving timing errors. Tables 2 and 3 summarize the positioning accuracy at 90% CDF considering different generalization factors for direct AI-based positioning and AI-based ToA prediction, respectively. Generally, once there is a change in these typical parameters, the positioning performance would degrade greatly and the trend in the direct AI-based positioning and AI-based ToA prediction is similar. Comparing the impact of these three factors, it is observed that the impact of receiving timing errors is the smallest due to the small range of timing errors. The impact of the network synchronization error is the most significant. There are two reasons, one reason is that the error range of the network synchronization is usually large. Another reason is all BSs may experience such errors. As for the impact of clutter parameters, more performance degradation is observed when the AI model is trained with clutter parameters of {60%, 6 m, 2 m} and tested by data set with clutter parameters of {40%, 2 m, 2 m}. The possible reason is the different LOS/NLOS conditions between the scenario with a clutter of {60%, 6 m, 2 m} and the scenario with a clutter parameter of {40%, 2 m, 2 m}. The scenario with clutter parameter of {60%, 6 m, 2 m} is full of NLOS-based propagation while lack of LOS-based propagation and the scenarios with clutter parameter of $\{40\%, 2m, 2m\}$ contain more LOS-based propagation. In this case, the trained AI model by data set with a clutter parameter of {60%, 6 m, 2 m} may not be able to extract the feature of LOS-based propagation fully. Then, inferior performance will be experienced when handling the LOS-based propagation in the scenario with a clutter parameter {40%, 2 m, 2 m}.
- To guarantee the generalization performance from the perspective of clutter parameter and receiving timing error, both optimizing the data set construction and using fine-tuning can be considered. Tables 4 and 5 summarize the positioning accuracy at 90% CDF for direct AI-based positioning and AI-based ToA prediction when mixed data sets and fine-tuning are used. For the impact of different clutter parameters, both optimizing the data set could provide less than 1 m positioning accuracy at 90% CDF. For the impact of receiving timing error, using mixed data set for training could improve the positioning accuracy at 90% CDF. For the positioning accuracy, while it seems that fine-tuning does not help much. The possible reason is that fewer samples with receiving timing errors are used for training

in the fine-tuning method. For the impact of network synchronization error, it is observed that using the mixed data set and fine-tuning could improve the positioning performance slightly but, on the hand, using these two methods still cannot meet the positioning accuracy of less than 1 m. In addition, when comparing these two generalization improvement solutions, it is observed that optimizing data set construction could achieve slightly better performance due to more samples from the target scenario.

• For network synchronization error, using mixed data sets and fine-tuning does not help much in improving the generalization capability. Other solutions can be considered. According to our test, when the network synchronization error is small, the positioning error would be small. For example, when the AI model is trained with the data set with mixed network synchronization error setting and tested with data set of 30 ns standard deviation, then the positioning error @90% for clutter parameter {40%, 2 m, 2 m} is reduced from around 3 m to less than 2 m, as shown in Figure 28. Thus, in the realistic work, solutions to reduce the network synchronization error could help to guarantee the performance. In addition, another approach is to use dedicated models for dedicated network synchronization error settings. For example, as shown in Figure 29, in the network with a synchronization error of 50 ns standard deviation, when the AI model is also trained with data set with synchronization error of 50 ns standard deviation, then better performance is also achieved. In this approach, the network should monitor the distribution of the network synchronization error, and once the distribution changes, then a model switch is needed. The details of the model switch could be one part of our future work.

Table 2. Positioning results summary of different affecting factors for direct AI-based positioning.

	Clutter Parameter	Positioning Error @90% CDF (m)
Impact from different clutter parameters from training	Training: {0.6,6,2} Test: {0.4,2,2}	7.0914
inpact noni unierent ciutier parameters nom training	Training {0.4,2,2} Test{0.6,6,2}	1.5328
Impact from network synchronization error	Training/Test: {0.6,6,2}	12.4486
inpact noin network synchronization error	Training/Test: {0.4,2,2}	14.5779
Impact from receiving error	Training/Test: {0.6,6,2}	1.7885
inpact noin receiving error	Training/Test: {0.4,2,2}	2.5998

Table 3. Positioning results summary of different affecting factors for AI-based ToA prediction.

	Clutter parameter	Positioning Error @90% CDF (m)
Impact from different clutter parameters from training	Training: {0.6,6,2} Test: {0.4,2,2}	7.1173
impact nom unerent clutter parameters nom training	Training {0.4,2,2} Test{0.6,6,2}	1.5413
Impact from notwork suppropriation arror	Training/Test: {0.6,6,2}	12.7748
inipact noin network synchronization error	Training/Test: {0.4,2,2}	15.4699
Impact from receiving timing error	Training/Test: {0.6,6,2}	1.3461
impact nom receiving timing error	Training/Test: {0.4,2,2}	2.2409



Figure 28. Performance comparison between different network synchronization errors.

	Table 4. Results summar	y of mixed data sets and	fine-tuning for direct	AI-based positioning
--	-------------------------	--------------------------	------------------------	----------------------

Simulation Cases		Positioning Error @90% CDF (m)	
	Clutter Parameter of Test Data Set	Mixed Parameter	Fine-Tuning
Different clutter parameters	{0.6,6,2}	0.5419	1.4315
Network synchronization error	{0.4,2,2} {0.6,6,2}	1.2736	4.7145
Network synchronization error	$\{0.4,2,2\}$	2.3022 0.8074	8.0254 1.6094
Receiving timing error	{0.4,2,2}	1.2604	2.1858

Table 5. Results summary of mixed data sets and fine-tuning for AI-based ToA prediction.

Simulation Cases		Positioning Error @90% CDF(m)	
	Clutter Parameter of Test Data Set	Mixed Parameter	Fine-Tuning
Different clutter parameters	{0.6,6,2}	0.6867	1.5325
	{0.4,2,2}	0.7974	0.8494
Network synchronization error	{0.6,6,2}	1.3461	4.4925
	{0.4,2,2}	2.2409	7.506
Receiving timing error	{0.6,6,2}	0.9341	1.6034
	{0.4,2,2}	1.2952	2.1152





6. Conclusions

In this paper, we investigate the generalization capability of AI-based positioning in IIoT scenarios by considering the typical features of IIoT. It is observed that the clutter distribution would impact the generalization capability. Yet using typical solutions, i.e., optimized training data sets or fine-tuning could relieve the side effect of the change in the clutter distributions. We further observe that the receiving timing error could also impact the generalization, but the impact is not so significant compared to other factors. Additionally, the traditional generalization improvement solutions also work for it. While for the network synchronization error, the impact on the generalization capability is obvious and the existing solutions, i.e., optimized data set construction or fine-tuning do not help much. Other solutions are needed. According to our test, we found that reducing the synchronization error in realistic networks could be one potential direction. In addition, using an AI model switch to adapt to the change in network synchronization error distribution could also be considered. The results of our investigation provide constructive guidance to develop AI models in realistic IIoT scenarios and also point out the potential challenges we need to cope with in future work.

Author Contributions: Conceptualization, Q.M.; methodology, Q.M.; software, H.Z.; validation, Q.M. and H.Z.; formal analysis, Q.M.; investigation, Q.M.; resources, M.W.; data curation, Q.M.; writing—original draft preparation, Q.M.; writing—review and editing, Q.M.; visualization, Q.M.; supervision, M.W.; project administration, Q.M. and M.W.; funding acquisition, M.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Laoudias, C.; Moreira, A.; Kim, S.; Lee, S.; Wirola, L.; Fischione, C. A survey of enabling technologies for network localization, tracking, and navigation. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3607–3644. [CrossRef]
- 2. TS 37.355; LTE Positioning Protocol (LPP). 3GPP Radio Access Network Working Group: Sophia Antipolis, France, 2022.
- 3. TR 38.855; Study on NR Positioning Support. 3GPP Radio Access Network Working Group: Sophia Antipolis, France, 2019.

- 4. TR 38.857; Study on NR Positioning Enhancements. 3GPP Radio Access Network Working Group: Sophia Antipolis, France, 2021.
- 5. Niitsoo, A.; Edelh, T.; Eberlein, E.; Hadaschik, N.; Mutschler, C. A deep learning approach to position estimation from channel impulse responses. *Sensors* **2019**, *19*, 1064. [CrossRef] [PubMed]
- Niitsoo, A.; Edelhäußer, T.; Mutschler, C. Convolutional Neural Networks for Position Estimation in TDoA-Based Locating Systems. In Proceedings of the 9th International Conference on Indoor Positioning and Indoor Navigation, Nantes, France, 24–27 September 2018; pp. 1–8.
- Martinez Sala, A.; Quir'os, R.; L'opez, E. Using neural networks and Active RFID for indoor location services. In Proceedings of the European Workshop Smart Objects: Systems, Technologies and Applications, Ciudad, Spain, 15–16 June 2010; pp. 1–9.
- 8. Singh, P.; Agrawal, S. TDOA Based Node Localization in WSN using Neural Networks. In Proceedings of the International Conference on Communication Systems and Network Technologies, Gwalior, India, 6–8 April 2013; pp. 400–404.
- 9. Chen, C.S. Artificial Neural Network for Location Estimation in Wireless Communication Systems. *Sensors* 2012, *12*, 2798–2817. [CrossRef] [PubMed]
- 10. Luo, J.; Gao, H. Deep Belief Networks for Fingerprinting Indoor Localization Using Ultrawideband Technology. *Int. J. Distrib. Sens. Netw.* **2016**, 2016, 18. [CrossRef]
- Le, D.V.; Meratnia, N.; Havinga, P.J.M. Unsupervised Deep Feature Learning to Reduce the Collection of Fingerprints for Indoor Localization Using Deep Belief Networks. In Proceedings of the 9th International Conference on Indoor Positioning and Indoor Navigation, Nantes, France, 24–27 September 2018; pp. 1–7.
- 12. Félix, G.; Siller, M.; Álvarez, E.N. A Fingerprinting Indoor Localization Algorithm based Deep Learning. In Proceedings of the 8th International Conference on Ubiquitous and Future Networks, Vienna, Austria, 5–8 July 2016; pp. 1006–1011.
- 13. Kim, K.S.; Lee, S.; Huang, K. A Scalable Deep Neural Network Architecture for Multi-Building and Multi-Floor Indoor Localization based on Wi-Fi Fingerprinting. *Big Data Anal.* **2018**, *4*, 1–17. [CrossRef]
- 14. Savic, V.; Larsson, E.G. Fingerprinting-Based Positioning in Distributed Massive MIMO Systems. In Proceedings of the 82nd IEEE Conference on Vehicular Technology, Boston, MA, USA, 6–9 September 2015; pp. 1–5.
- 15. Kuo, R.; Tseng, W.; Tien, F.; Liao, W. Application of an Artificial Immune System-based Fuzzy Neural Network to a RFID-based Positioning System. *J. Comput. Ind. Eng.* **2012**, *63*, 943–956. [CrossRef]
- Vaghefi, S.Y.M.; Vaghefi, R.M. A Novel Multilayer Network Model for TOA-Based Localization in Wireless Sensor Networks. In Proceedings of the International Joint Conference on Neural Networks, San Jose, CA, USA, 31 July–5 August 2011; pp. 3079–3084.
- Lewandowski, A.; Köster, V.; Wietfeld, C.; Michaelis, S. Support Vector Machines for Non-Linear Radio Fingerprint Recognition in Real-Life Industrial Environments. In Proceedings of the International Conference on Technical Meeting, San Diego, CA, USA, 24–26 January 2011; pp. 628–634.
- 18. Akram, B.A.; Akbar, A.H.; Shafiq, O. HybLoc: Hybrid IndoorWi-Fi Localization Using Soft Clustering-Based Random Decision Forest Ensembles. *IEEE Access* 2018, *6*, 38251–38272. [CrossRef]
- Mascharka, D.; Manley, E. LIPS: Learning Based Indoor Positioning System using mobile phone-based sensors. In Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference, Las Vegas, NV, USA, 9–12 January 2016; pp. 968–971.
- Iqbal, Z.; Luo, D.; Henry, P.; Kazemifar, S.; Rozario, T.; Yan, Y.; Westover, K.; Lu, W.; Nguyen, D.; Long, T.; et al. Accurate Real Time Localization Tracking in a Clinical Environment using Bluetooth Low Energy and Deep Learning. *PLoS ONE* 2017, 13, e0205392. [CrossRef] [PubMed]
- Ibrahim, M.; Torki, M.; ElNainay, M. CNN based Indoor Localization using RSS Time-Series. In Proceedings of the 2018 IEEE Symposium on Computers and Communications, Natal, Brazil, 25–28 June 2018; pp. 1044–1049.
- 22. Sahar, A.; Han, D. An LSTM-based Indoor Positioning Method Using Wi-Fi Signals. In Proceedings of the 2nd International Conference on Vision, Image and Signal Processing, Las Vegas, NV, USA, 27–29 August 2018; pp. 43:1–43:5.
- Feigl, T.; Nowak, T.; Philippsen, M.; Edelhäußer, T.; Mutschler, C. Recurrent Neural Networks on Drifting Time-of-Flight Measurements. In Proceedings of the 9th International Conference on Indoor Positioning and Indoor Navigation, Nantes, France, 24–27 September 2018; pp. 206–212.
- 24. Mohammadi, M.; Al-Fuqaha, A.; Guizani, M.; Oh, J.S. Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services. *IEEE Internet Things J.* 2018, *5*, 624–635. [CrossRef]
- 25. What Is Generalization in Machine Learning? Available online: https://deepai.space/what-is-generalization-in-machine-learning (accessed on 15 March 2023).
- 26. de Sousa, M.N.; Thomä, R.S. Enhancement of localization systems in nlos urban scenario with multipath ray tracing fingerprints and machine learning. *Sensors* **2018**, *18*, 4073. [CrossRef] [PubMed]
- Arnold, M.; Dorner, S.; Cammerer, S.; Brink, S.T. On deep learning-based massive mimo indoor user localization. In Proceedings of the (SPAWC 2022) 2022 IEEE International Workshop on Signal Processing Advances in Wireless Communications, Oulu, Finland, 4–6 July 2022; pp. 1–5.
- De Bast, S.; Guevara, A.P.; Pollin, S. Csi-based positioning in massive mimo systems using convolutional neural networks. In Proceedings of the Vehicular Technology Conference, Virtual, 18 November–16 December 2020; IEEE: New York, NY, USA, 2020; pp. 1–5.

- 29. Stahlke, M.; Feigl, T.; Castañeda García, M.H.C.; Stirling-Gallacher, R.A.; Seitz, J.; Mutschler, C. Transfer Learning to adapt 5G AI-based Fingerprint Localization across Environments. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference, Helsinki, Finland, 19–22 June 2022.
- 30. *TR 38.901;* Study on Channel Model for Frequencies from 0.5 to 100 GHz. 3GPP Radio Access Network Working Group: Sophia Antipolis, France, 2021.
- He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.