3GPP TSG RAN WG1 #112bis-e **R1-2302481**

e-Meeting, April 17th – April 26th, 2023

Source: **vivo**

Title: **Evaluation on AI/ML for positioning accuracy enhancement**

Agenda Item: **9.2.4.1**

Document for: **Discussion and Decision**

1. Introduction

Many positioning methods have been specified in Rel-16 and Rel-17 NR positioning, to obtain position estimation with target horizontal positioning accuracies of <0.2 m (90%) for IIoT use cases and <1 m (90%) for commercial use cases. However, the performance of these positioning methods highly relies on the existence of multiple LOS (line-of-sight) paths between the target terminal and multiple TRPs (Transmission-Reception Points). In the scenarios with extremely low LOS probability, positioning accuracy would decrease dramatically, which may be not able to satisfy the high-accuracy positioning requirements stemming from new applications and industry verticals.

The AI/ML technology has powerful abilities in feature extraction, environment awareness, complex problem modeling and processing. In recent years, applying AI/ML into air-interface has attracted great attentions from academics to industries, and a lot of meaningful exploration has been made to verify the performance gain compared to conventional non-AL/ML schemes. Related research has also verified that the AI/ML technology has the potential to significantly improve the performance of wireless communications.

Under this background, a new SI on *Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface* has been agreed at RAN #94e[1], including three use cases to assess the applications of AI/ML in air-interface. Among them, AI/ML based positioning accuracy enhancement is included, with the target to improve the positioning accuracy for different scenarios, especially for some challenging scenarios with heavy NLOS (non-line-of-sight) conditions.

The objective of the new SI for RAN1 AI/ML based positioning includes the following:

> Study the 3GPP framework for AI/ML for air-interface corresponding to each target use case regarding aspects such as performance, complexity, and potential specification impact.
> Use cases to focus on:
> 1. Initial set of use cases includes:
> b. Positioning accuracy enhancements for different scenarios including, e.g., those with heavy NLOS conditions [RAN1]

All agreements reached in previous meetings can refer to Appendix C. In this contribution, we present our simulation results and observations to demonstrate the performance gain of applying AI/ML technology onto positioning for various scenarios.

2. Evaluation scenarios and methodology

According to the SID [1], the evaluation methodology should be based on statistical models (from TR 38.901 and TR 38.857 [positioning]), for link and system level simulations. In TR38.901, multiple InF scenarios are defined, focusing on factory halls with varying sizes and varying levels of clutter density. The InF scenarios include:

- **InF-SL** Indoor Factory with Sparse clutter and Low base station height (both Tx and Rx are below the average height of the clutter)
- **InF-DL** Indoor Factory with Dense clutter and Low base station height (both Tx and Rx are below the average height of the clutter)
- **InF-SH** Indoor Factory with Sparse clutter and High base station height (Tx or Rx elevated above the clutter)
- **InF-DH** Indoor Factory with Dense clutter and High base station height (Tx or Rx elevated above the clutter)

- **InF-HH** Indoor Factory with High Tx and High Rx (both elevated above the clutter)

Among them, the DH scenario with clutter parameter {density: 60%, height: 6m, size: 2m} have extremely low LOS probability (95% NLOS links, as shown in Figure 1) and it is challenging to achieve accurate position estimation by utilizing the conventional RAT-dependent positioning methods, such as TDoA, RTT and so on. Due to the dramatic different distributions of LOS/NLOS path in different InF scenarios, we think an AI/ML model trained on dataset from a single InF scenario cannot guarantee its performance when the actual deployment scenario is not a perfect match of the scenario where the trained dataset coming from. Therefore, we think it's essential to evaluate AI/ML model performance under different settings and scenarios to test and verify its' effective performance.

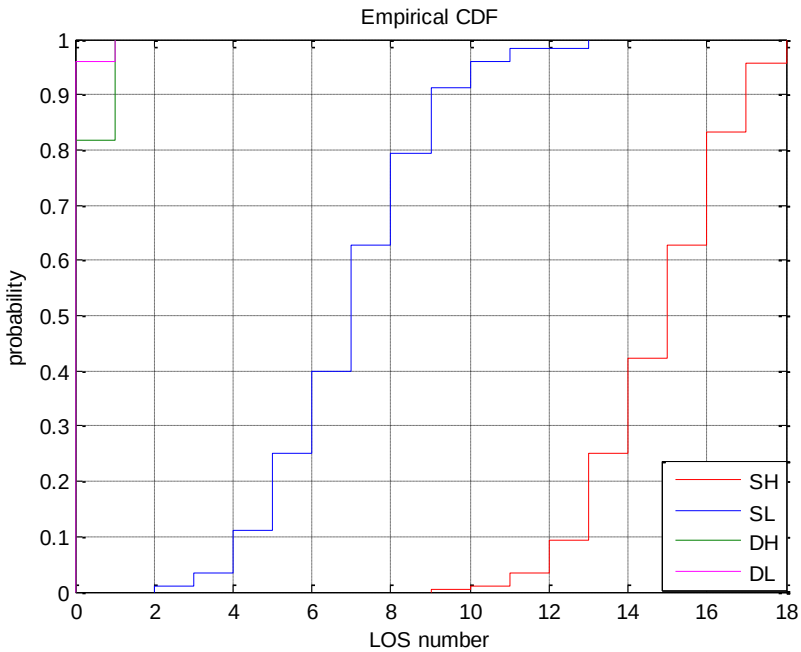


Figure 1 LOS probability of 4 InF scenarios (SL, DL, SH, DH)

Generalization is one of the key issues for all AI/ML based applications, and AI/ML based positioning is of no exception. The generalization capability of AI/ML model may be affected by the structure of AI/ML model, the variety of training data set and the training strategy. It is better to keep the training loss to be an accurate approximation of the generalization loss uniformly for all hypotheses. When performing evaluation of performance related KPIs, generalization performance should be seriously considered, and different levels of generalization may need to be verified. For example, whether the performance maintains when AI model transfers from one cell to another, from one drop to another, or from one scenario to another.

3. Evaluation results of sub use cases

At the RAN1 #110 meeting, it was agreed that:

Agreement

For AI/ML-based positioning, both approaches below are studied and evaluated by RAN1:

- Direct AI/ML positioning
- AI/ML assisted positioning

In this section, we provide our simulation results of basic performance evaluation for two representative sub use cases of AI/ML based positioning. The datasets with spatial consistency, including training dataset, validation dataset and test dataset, are generated with system-level simulation platform to train, validate and test AI/ML model, respectively. The details are reported in each sub-section below. Common parameter assumptions for scenarios are provided in Appendix A, and details about AI/ML model training/validation and testing parameters are provided in Appendix B.

3.1. Direct AI/ML positioning

For direct AI/ML positioning, UE position can be directly estimated according to multiple TRPs' Channel Impulse Response (CIR) vectors, as shown in Figure 2. Note that, AI/ML model can be deployed at the UE side or network side.

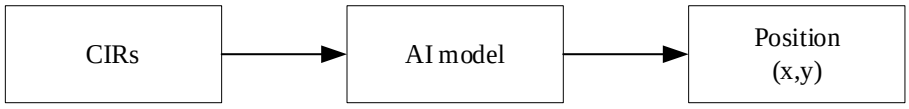


Figure 2 Direct AI/ML positioning with multiple TRPs' CIRs

The InF-DH scenario with size 120m × 60m and clutter density {0.6, 6, 2} is adopted for evaluation. For each UE, we generate time-domain channel response data points (with the dimension of $4\times32\times4096\times18$) labeled with associated location by system level simulation platform [3]. Then, we sample by truncating the first 256 time-domain points based on the 1st Tx antenna element and the 1st Rx antenna element from CIR. Finally, the sampled CIR is reshaped into the dimension of $256\times1\times18$ as the input of AI/ML model. Moreover, 25k samples are used to train the adopted Vision Transformer model [4], and 1k samples are used for testing.

3.1.1. Performance comparison with baselines

The conventional positioning methods in previous releases are considered as baselines. From the simulation results in Table 1, it is observed that the positioning errors of baselines are larger than 20m due to the low probability of LOS path, which is not able to satisfy the requirements of high accuracy positioning in heavy NLOS scenarios. While AI technology can significantly improve positioning accuracy and reaps a conspicuous performance gain (<1m @90%). Thus, we expect that AI technology can be exploited to significantly improve the positioning accuracy, especially for heavy NLOS scenarios.

Table 1 CDF of positioning accuracy (m) of different positioning methods

Scenario	Positioning methods	50%	67%	80%	90%
InF-DH {0.6,6,2}	DL-TDOA	8.38	11.09	15.95	32.12
	UL-TDOA	8.60	11.52	16.33	32.81
	RTT	8.32	11.42	15.72	32.41
	AOA	8.13	10.36	14.09	20.16
	Machine learning	0.35	0.49	0.70	0.99

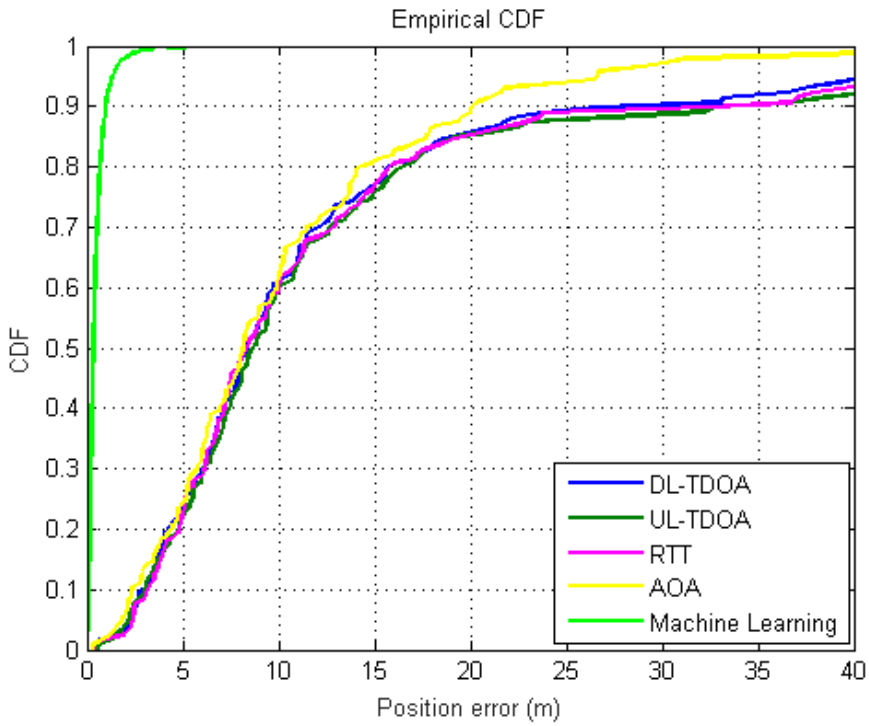


Figure 3 CDF of positioning accuracy (m) of different positioning methods

Observation 1: AI/ML based positioning can significantly improve the positioning accuracy compared to existing RAT-dependent positioning methods in heavy NLOS scenarios.

3.1.2. Model input

At the RAN1 #111 meeting, it was agreed that:

Agreement

For reporting the model input dimension N_{TRP} * N_{port} * N_t of CIR and PDP, N_t refers to ~~the first~~ N_t consecutive time domain samples.

- If N'_t ($N'_t < N_t$) samples with the strongest power are selected as model input, with remaining ($N_t - N'_t$) time domain samples set to zero, then companies report value N'_t in addition to N_t. It is also assumed that timing info for the N'_t samples need to be provided as model input.

Agreement

For reporting the model input dimension N_{TRP} * N_{port} * N_t:

- If the model input is CIR, then each input value of CIR is a complex number, i.e. it contains two real values, either {real, imaginary} or {magnitude, phase}.
- If the model input is PDP, then each input value of PDP is a real value.

Accordingly, CIR with dimension $18\times1\times256$ (N_{TRP} * N_{port} * N_t) is adopted as model input in our simulations. Simulation comparison of different input selection for AI/ML based positioning is shown in Table 2. We can see that time domain channel CIR as the input of AI/ML model can obtain the best positioning accuracy compared to other inputs, such as power, delay and angle of the first path. The reason we believe is that original CIR contains richer features which may be strongly related to the target UE's location. In this sense, AI/ML model can be regarded as a feature extractor, capturing location related features from CIR in an implicit manner, and then determining the location according to these features.

Table 2 Evaluation results of different model inputs for AI/ML model deployed on UE or Network side, without model generalization, ViT, UE distribution area = [120x60 m]

Model input	Model output	Label	Clutter param	Dataset size	AI/ML complexity	Horizontal positioning accuracy at CDF=90% (meters)

				Train	Test	Model complexity	Computational complexity	AI/ML
CIR	Pos.	0	{0.6,6,2}	25k	1k	1.65M	22.30M	0.99
Power + delay + angle of the first path	Pos.	0	{0.6,6,2}	25k	1k	1.65M	22.30M	1.19
Power + delay of the first path	Pos.	0	{0.6,6,2}	25k	1k	1.65M	22.30M	1.31
Delay + angle of the first path	Pos.	0	{0.6,6,2}	25k	1k	1.65M	22.30M	1.43
Angle + power of the first path	Pos.	0	{0.6,6,2}	25k	1k	1.65M	22.30M	1.79

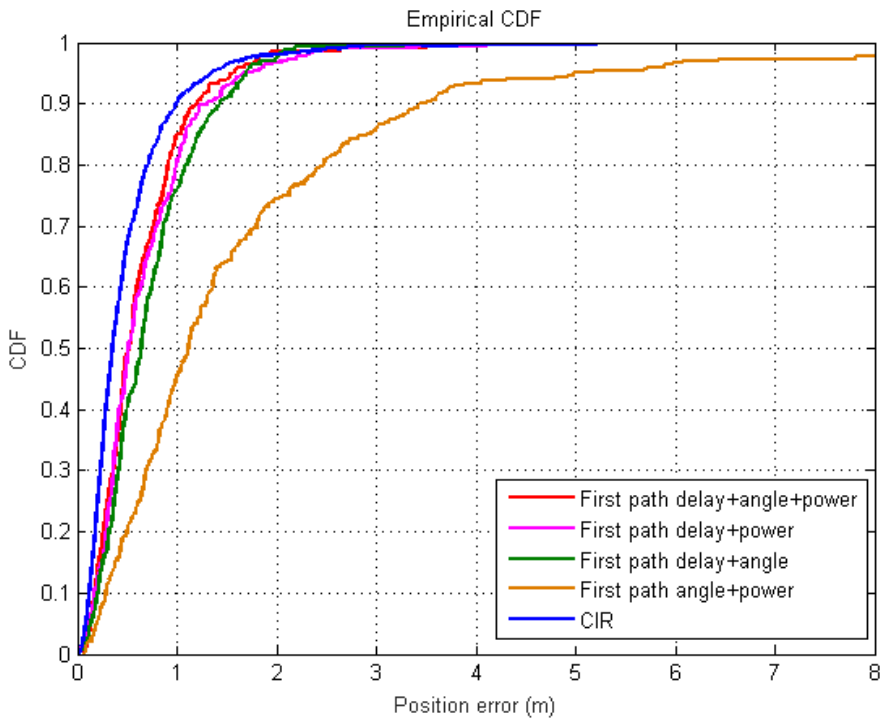


Figure 4 CDF of positioning accuracy (m) of different measurements

Observation 2: Different inputs of AI/ML model will affect the positioning performance for AI/ML based positioning. Time domain channel CIR as the input of AI/ML model obtains the best positioning accuracy.

Proposal 1: Capture in the TR that time domain CIR as the model input for direct AI/ML positioning obtains the best performance compared to other model inputs.

Proposal 2: Support time domain CIR as the model input at least for direct AI/ML positioning.

3.2. AI/ML assisted positioning

At the RAN1 #110b-e meeting, it was agreed that:

Agreement

For evaluation of AI/ML assisted positioning, the following intermediate performance metrics are used:

- LOS classification accuracy, if the model output includes LOS/NLOS indicator of hard values, where the

LOS/NLOS indicator is generated for a link between UE and TRP;

- Timing estimation accuracy (expressed in meters), if the model output includes timing estimation (e.g., ToA, RSTD).
- Angle estimation accuracy (in degrees), if the model output includes angle estimation (e.g., AoA, AoD).
- Companies provide info on how LOS classification accuracy and timing/angle estimation accuracy are estimated, if the ML output is a soft value that represents a probability distribution (e.g., probability of LOS, probability of timing, probability of angle, mean and variance of timing/angle, etc.)

For AI/ML assisted positioning, AI/ML technology is utilized to extract some intermediate features from channel state information (e.g., CIR), such as TOA, LOS/NLOS identification, and so on. Specifically, as shown in Figure 5, instead of constructing an AI/ML model with 18 TRPs' CIRs as input and the target UE's location as output, we consider a more general framework with one TRP's CIR as the input and an intermediate feature (such as TOA of that TRP at the target UE) as the output for each TRP, respectively. Based on the intermediate feature extracted from CIR of each TRP, the location of the target UE can be further derived by utilizing other positioning algorithms, including AI-based or non-AI based algorithms. In order to distinguish from aforementioned *direct AI/ML positioning* method based on multi-TPRs' CIRs, we call it *AI/ML assisted positioning*, i.e., CIR-intermediate feature-positioning.

From our views, we mainly focus on single-TRP construction and the model of each TRP shares the same model structure but varying model parameters, and other agreed constructions are also evaluated as presented in section 3.2.1. Optionally, it is also possible to construct a common model trained with all TRPs' data. The main motivation comes from our considerations about AI/ML model generalization and practical deployment in real environment for AI based positioning. The AI/ML model related with multiple TRPs is strongly correlated with TRPs' distribution, and may not work well once TPRs' distribution changes, such as the number of TPRs, the location of each TRP. Apparently, an AI model trained with multiple TRPs' CIRs works the best in those trained scenarios with multiple TRPs, which in turn means that large number of field data needs to be collected from real deployment and computation & time-consuming model training/validation process needs to be conducted from scratch for each scenario. However, AI/ML assisted positioning method estimating intermediate feature from single-TRP's CIR is independent of these factors, and can be largely compatible with existing positioning protocol framework (i.e., LPP) specified in previous releases.

In this section, we mainly focus on the evaluations of two typical schemes, i.e., AI/ML based TOA estimation and AI/ML based LOS/NLOS identification, and further analyze their pros and cons.

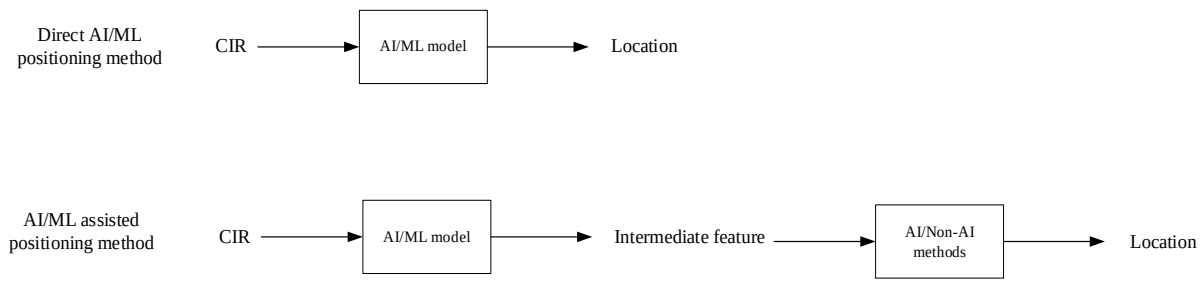


Figure 5 The framework of AI/ML assisted positioning method

3.2.1. AI/ML based TOA estimation

At the RAN1 #111 meeting, it was agreed that:

Agreement

For AI/ML assisted positioning, evaluate the three constructions:

- Single-TRP, same model for N TRPs
- Single-TRP, N models for N TRPs
- Multi-TRP (i.e., one model for N TRPs)

Note: Individual company may evaluate one or more of the three constructions.

Three constructions for AI/ML assisted positioning are agreed in the RAN1 #111 meeting. In the following, we comprehensively evaluate their performance. Specifically, we evaluate the performance of AI/ML based TOA estimation where TOA from a TRP to a target UE is taken as the intermediate feature. The specific procedures are presented as follows. Firstly, we obtain the input of the AI/ML model with the dimension of $256\times1\times18$ in a similar manner as described in the section 3.1. It is further divided into 18 vectors each with dimension of 256×1 as the input of single-TRP's model for Construction 1 and Construction 2, while full-dimension CIR is input without

splitting for Construction 3. Based on the relative location of UEs and each TRP, AI/ML model(s) for TOA estimation associated with straight-line (LOS) distance can be trained. Then, TOA associated with each TRP can be estimated according to the trained AI/ML models with UE's CIR as the input.

According to estimated TOAs associated with multiple TRPs, we adopt the CHAN positioning algorithm [12] to estimate UE's location. Specifically, we firstly select four TOAs with the highest accuracy from multiple TOA estimations when assuming that TOA errors can be obtained. In practice, conventional TRP selection algorithms can also be used to select the TOAs with minimal errors, such as Receive Autonomous Integrity Monitoring algorithm. Based on the selected TOAs and prior locations of TRPs, the final location of the target UE is calculated by CHAN algorithm.

3.2.1.1. Construction 1: Single-TRP, N models for N TRPs

As shown in Table 3, for Construction 1 (Single-TRP, N models for N TRPs), the AI/ML based TOA estimation positioning method (0.73m@90%) achieves remarkable performance gain compared to direct AI/ML positioning method (0.99m@90%).

Table 3 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	Drop1	Drop1	25k	1k	1.65M	22.30M	0.99
CIR	TOA	0	Drop1	Drop1	25k	1k	4.26M*18	8.50M*18	0.73

Table 4 CDF of estimation accuracy of intermediate feature TOA (meter) , UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Drop1	Drop1	25k	1k	4.26M*18	8.50M*18	0.62

According to the above agreement, the intermediate performance of single-TRP TOA estimation is also presented in Table 4. For convenience, the unit of TOA is set to meter ($3\times10^8\times(T_r-T_t)$), where T_r and T_t denote time of arrival and time of departure of the target signal when assuming that LOS path is exist, respectively. Particularly, @90% CDF is not our concern for the evaluation of intermediate performance since only 4 TOAs are required for TOA based positioning but not all TRPs' TOAs. Thus, we adopt the accuracy of the highest fourth TOA (TOP-4th) as a performance metric of TOA estimation. For example, TOP-4th TOA accuracy can be obtained by conventional TRP selection algorithms or going through all combinations of TOAs.

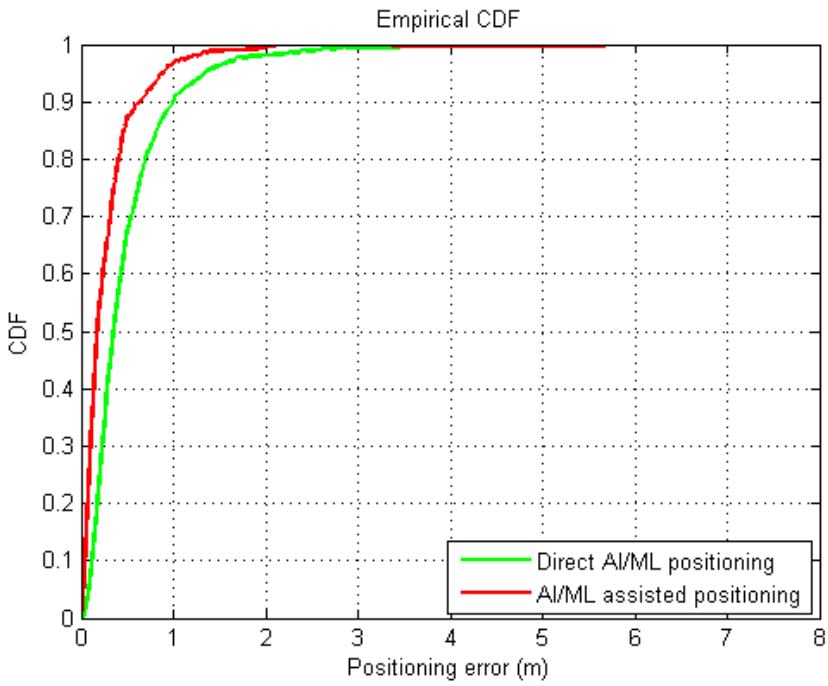


Figure 6 CDF of positioning accuracy

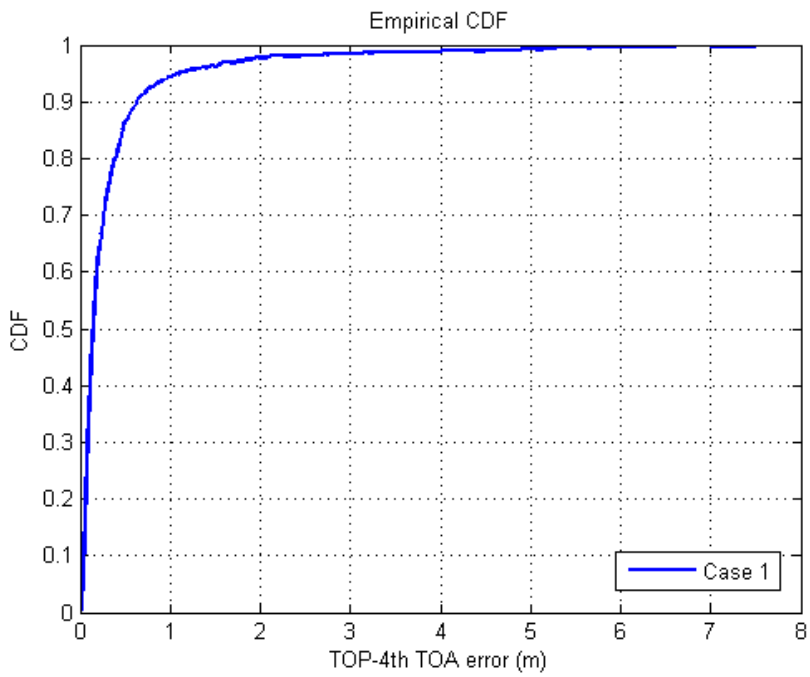


Figure 7 CDF of TOA estimation accuracy

Observation 3: For Construction1 (Single-TRP, N models for N TRPs), the AI/ML based TOA estimation positioning method achieves remarkable performance gain compared to direct AI/ML positioning method.

3.2.1.2. Construction 2: Single-TRP, same model for N TRPs

As shown in Table 5, when model input contains both CIR and related TRP's information (e.g., TRP's ID) , AI/ML based TOA estimation positioning method (0.83m@90%) achieves higher positioning performance compared to direct AI/ML positioning method (0.99m@90%). However, when only CIR without related TRP's information is input into the AI/ML model, obvious positioning accuracy degradation is observed. The reason behind is that it is difficult to achieve such regression (TOA estimation is a regression problem) when the same output (TOA) is related to greatly different inputs (CIRs related to different TRPs). Incorporating TRP's information into model input can facilitate the

AI/ML model distinguish which TRP the CIR is associated with. This performance gain may be also achieved by using a more complex model and more training data.

Table 5 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR w. BS info.	TOA	0	Drop1	Drop1	25k*18	1k*18	11.92M*1	23.79M*1	0.83
CIR w/o. BS info.	TOA	0	Drop1	Drop1	25k*18	1k*18	11.92M*1	23.79M *1	2.57

Table 6 CDF of estimation accuracy of intermediate feature TOA (meter) , UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m) @90% CDF=90%**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR w. BS info.	TOA	0	Drop1	Drop1	25k*18	1k*18	11.92M *1	23.79M *1	0.76
CIR w/o. BS info.	TOA	0	Drop1	Drop1	25k*18	1k*18	11.92M*1	23.79M *1	2.35

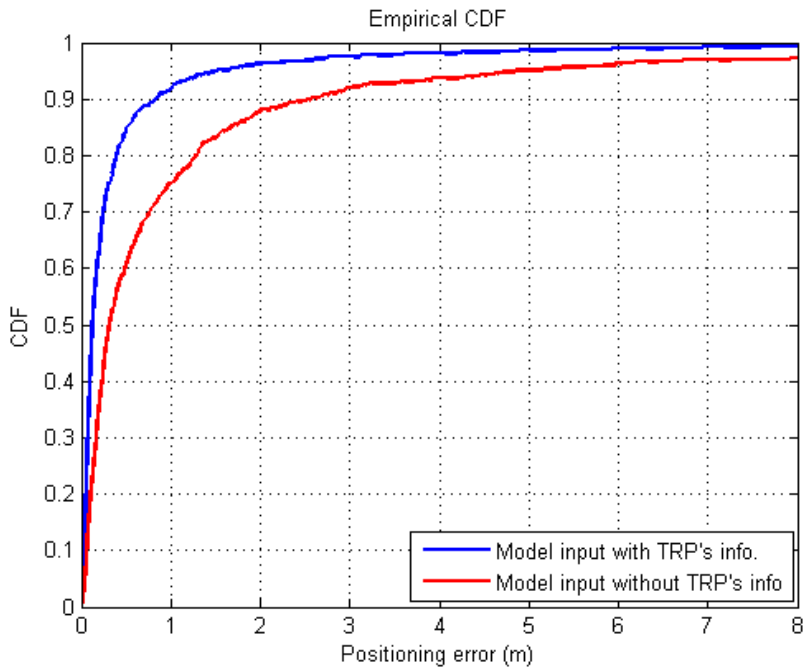


Figure 8 CDF of positioning accuracy

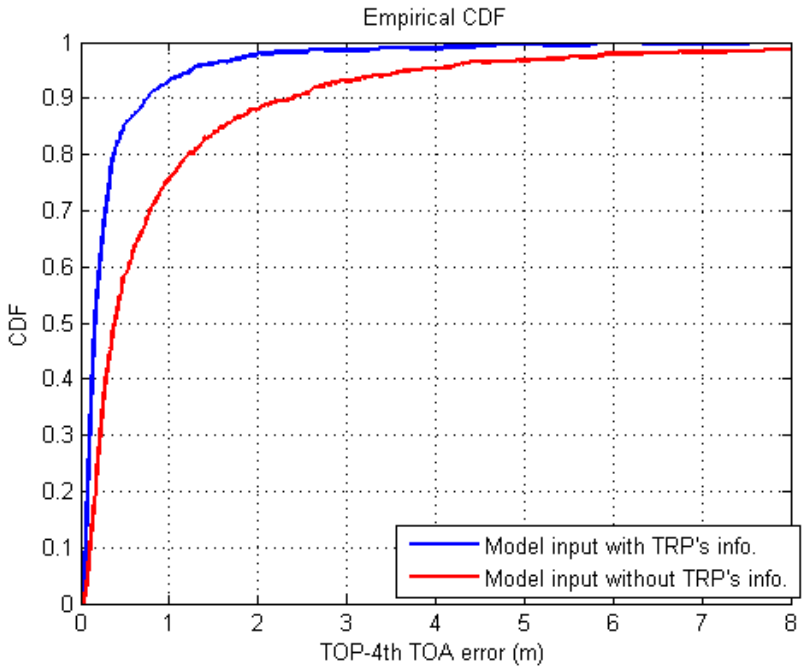


Figure 9 CDF of TOA estimation accuracy

Observation 4: For Construction 2 (Single-TRP, same model for N TRPs), it is beneficial to incorporate TRP's information into model input so as to improve the positioning accuracy.

3.2.1.3. Construction 3: Multi-TRP, one model for N TRPs

As shown in Table 7, we comprehensively compare three Constructions for AI/ML assisted positioning. The simulation results illustrate that Construction 1 reaps the best positioning accuracy at the cost of higher complexity. Moreover, Construction 3 has the lowest positioning accuracy despite its highest TOA estimation accuracy since selected TRPs with the high TOA estimation accuracy have a high probability of being located in a straight line.

Table 7 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Construction	**Model input**	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
Construction 1	CIR	TOA	0	Drop1	Drop1	25k	1k	4.26M*18	8.50M*18	0.73
Construction 2	CIR	TOA	0	Drop1	Drop1	25k*18	1k*18	11.92M*1	23.79M *1	0.83
Construction 3	CIR	TOA	0	Drop1	Drop1	25k	1k	1.65M	22.30M	1.08

Table 8 CDF of estimation accuracy of intermediate feature TOA (meter) , UE distribution area = [120x60 m]

Construction	**Model input**	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m) @90% CDF=90 %**
Construction 1	CIR	TOA	0	Drop 1	Drop 1	25k	1k	4.26M*18	8.50M*18	0.62

Construction 2	CIR	TOA	0	Drop 1	Drop 1	25k*18	1k*18	11.92M*1	23.79M*1	0.76
Construction 3	CIR	TOA	0	Drop 1	Drop 1	25k	1k	1.65M	22.30M	0.52

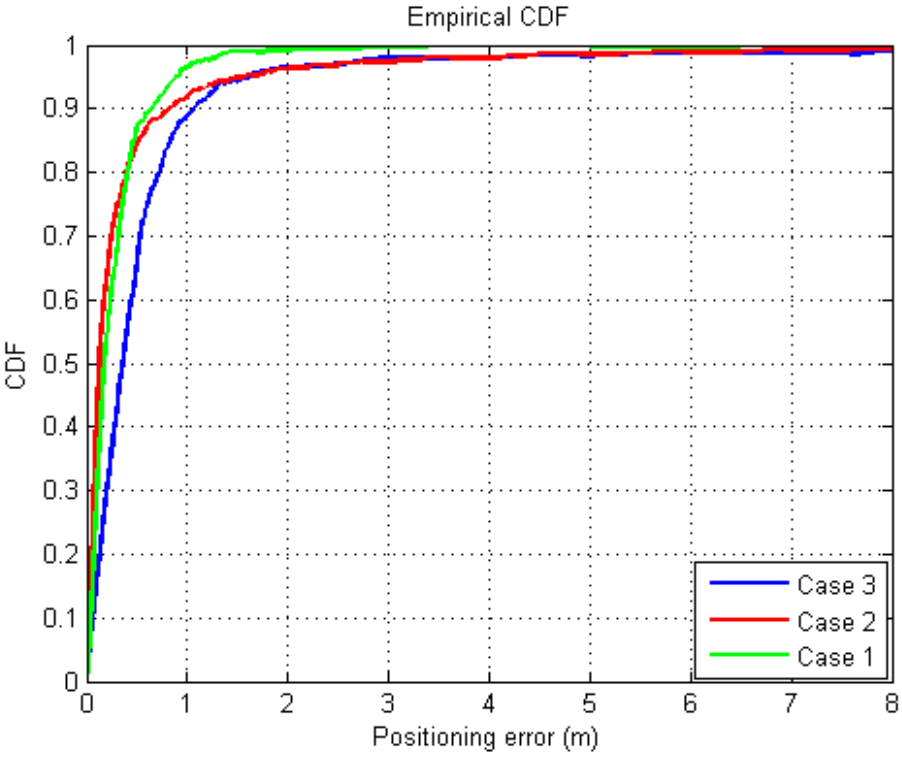


Figure 10 CDF of positioning accuracy

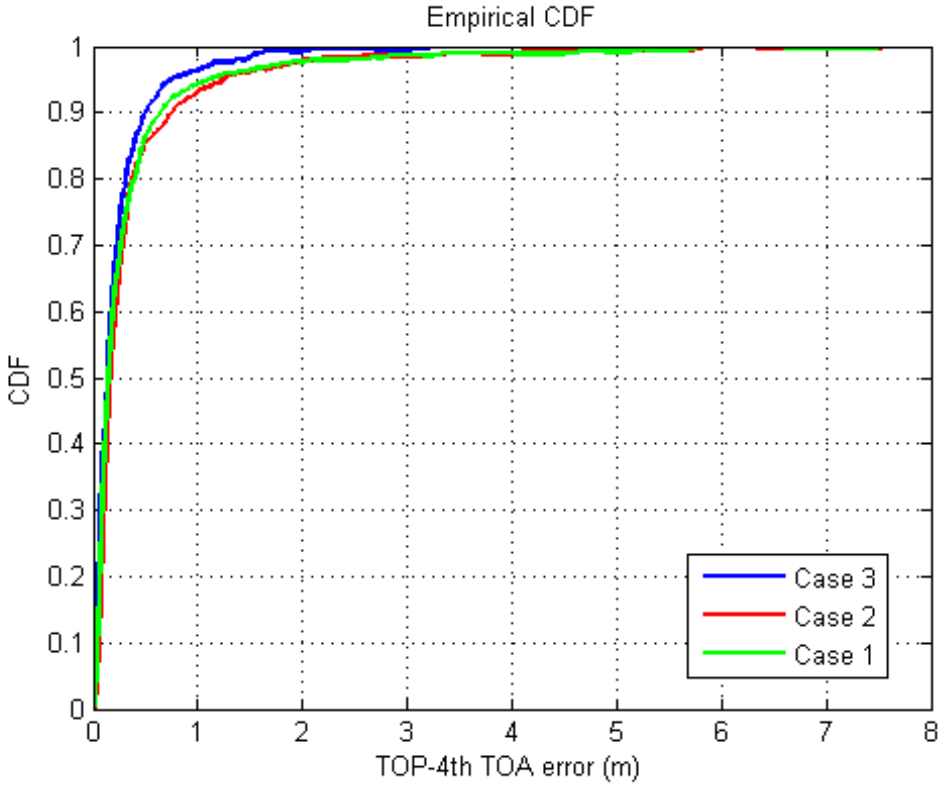


Figure 11 CDF of TOA estimation accuracy

3.2.2. Discussion on intermediate measurement, TOA or RSTD?

RSTD reporting for positioning has been supported in legacy specifications to eliminate the impact of timing error. For AI/ML based TOA estimation, there is no doubt that UE side can calculate RSTD when reference TRP is given and then reports RSTD to LMF side, and in this way AI/ML assisted positioning can be achieved while without any specification impact. However, from the perspective of positioning performance, we think TOA reporting may be benefit. In the following, we analyze the basic principle and evaluate their positioning performance for TOA based positioning and RSTD based positioning.

3.2.2.1. Basic principle

TOA based positioning is a positioning method that estimates an unknown target location based on a propagation time of a radio signal from the source to the target. The geometrical interpretation of the TOA based positioning method is illustrated in Figure 12. The distance (TOA) between the ith source node and the target node is given by

$$r_i = \|\mathbf{s}_i - \mathbf{u}\|_2 = \sqrt{(x_i - x)^2 + (y_i - y)^2}\ .$$

This is a circular equation in which a possible solution exists on a circle centered at the corresponding source node. As the measurement at each source node provides a circle with different centers and radius determined by r_i, the target location can be found at the point where all circles intersect. In case of TOA-based localization in two-dimensional space, at least three source nodes are required to obtain a unique solution of a target location.

The positioning accuracy of TOA based positioning is positively related to the accuracy of TOA estimation. In general, the accurate estimation of TOA requires a clock synchronization between the source nodes and the target node, including UE timing error and network synchronization error.

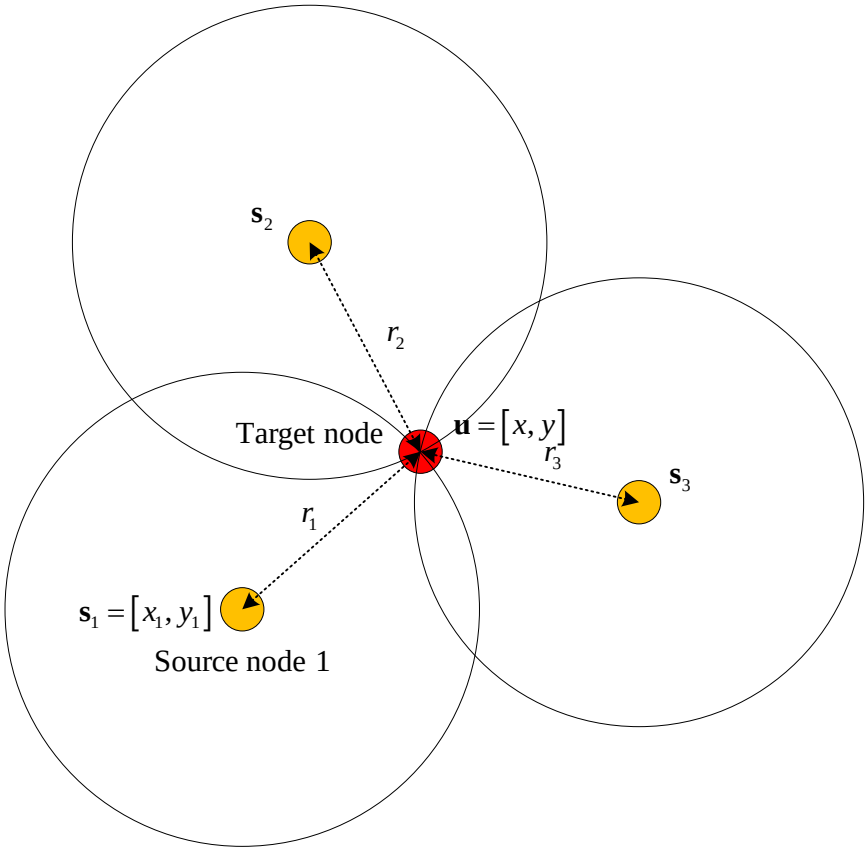


Figure 12 Illustration of TOA based positioning method

To mitigate the impact of clock synchronization especially for UE timing error, RSTD based positioning is adopted. The geometrical interpretation of the RSTD based positioning method is illustrated in Figure 13, in which the curves of same color represent a hyperbola generated by a hyperbolic equation, with the TRPs' locations as the

foci. Assuming the first source node is a reference node, subtracting the TOA measurement at the reference from the measurement at the ith source node yields the RSTD measurement, which is given by

$$r_{i,1} = r_i - r_1 = \sqrt{(x_i - x)^2 + (y_i - y)^2} - \sqrt{(x_1 - x)^2 + (y_1 - y)^2}$$

This is a hyperbolic equation in which a possible solution lies on a hyperbola. In case of RSTD-based localization in two-dimensional space, at least four source nodes are required to obtain a unique solution of a target location since there exists two intersections of the hyperbole.

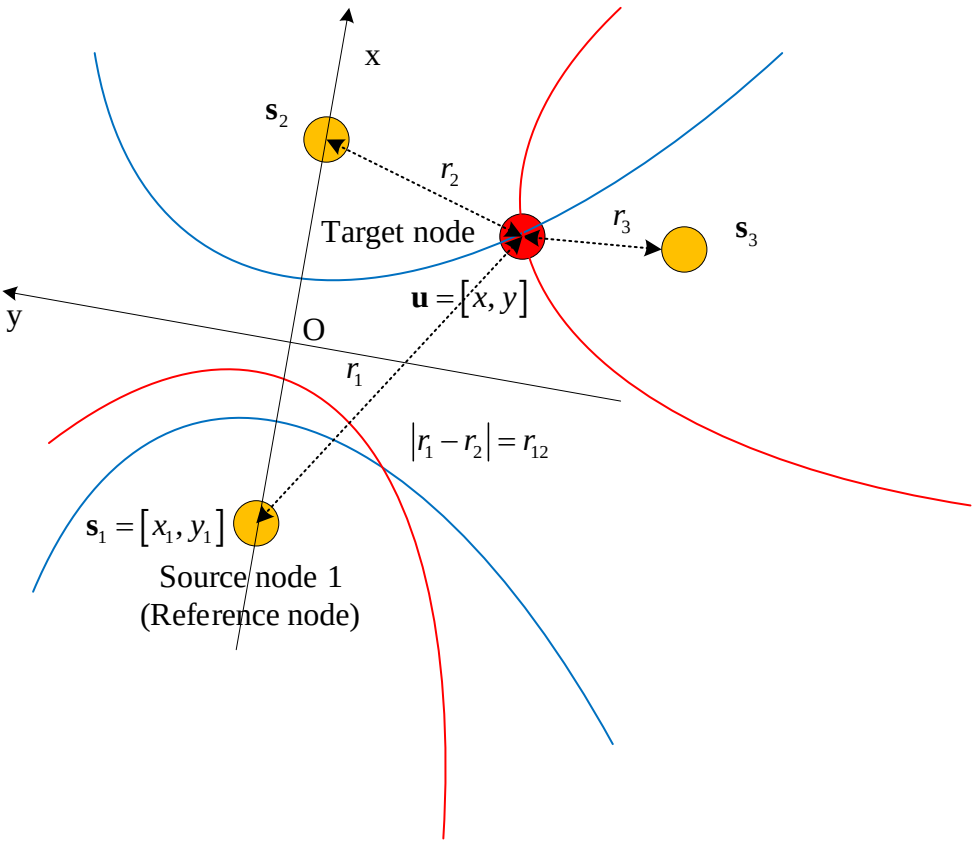


Figure 13 Illustration of RSTD based positioning method

3.2.2.2. Evaluation and Analysis

From section 3.2.2.1, it is worth noting that since the RSTD measurements forming the hyperbole are derived with respect to the TOA measurement at the reference source node, all RSTD measurements and their hyperboles are affected by the two TOA errors, i.e., TOA errors at the reference node and the target node, and thereby potentially making the RSTD based positioning less accurate than the TOA based positioning. By utilizing the CHAN algorithm [12], we evaluate their positioning performance of TOA based positioning and RSTD based positioning.

As shown in Table 9, the positioning performance of TOA based positioning and RSTD based positioning is evaluated under different TRP number configurations, and we have the following observations:

- The positioning accuracy of TOA based positioning is greatly better than that of RSTD based positioning regardless of the number of TRPs used for positioning calculation.
- With the increase of TRP number for location calculation, the positioning accuracy is obviously improved from 9.03m@90% to 2.04m@90% for RSTD based positioning, which confirms that RSTD based positioning is more sensitive to the number of TRP than TOA based positioning.

As shown in Table 10, simulation results show that RSTD reporting is not helpful for mitigating the impact of UE timing error for AI assisted positioning in the DH{0.6, 6, 2} scenarios, whose positioning accuracy is severely worse than TOA based positioning. The reason behind is that RSTD based positioning can eliminate the impact of timing error only when the reference TRP and the target TRP shares the same TOA estimation error. In our evaluation, the TOAs used to calculate RSTD are estimated by AI/ML models (Construction 1: Single-TRP, N models for N TRPs). Even if the same UE measures CIRs of the reference TRP and the target TRP, the estimated TOA errors for the reference TRP and the target TRP are greatly different since AI/ML based TOA estimation dependents on the CIR fingerprint but not first-path delay for heavy-NLOS scenarios.

Furthermore, TOA contains more raw positioning-related information compared to RSTD. Consequently, TOA reporting may provide more flexibility in positioning implementation, such as reporting TOA with some soft information. Additionally, reporting TOA is expected to fully maximize the potential of AI/ML assisted positioning, ultimately leading to improved positioning performance.

Table 9 Evaluation results for TOA based positioning and RSTD based positioning, UE distribution area = [120x60 m]

Measurement for location calculation	**UE timing error**	**Model input**	**Model output**	**Number of TRP for location calculation**	**Positioning accuracy(m) @90% CDF=90%**
TOA	0ns	CIR	TOA	4	0.62
RSTD	0ns	CIR	TOA	4	9.03
TOA	0ns	CIR	TOA	6	0.64
RSTD	0ns	CIR	TOA	6	2.04

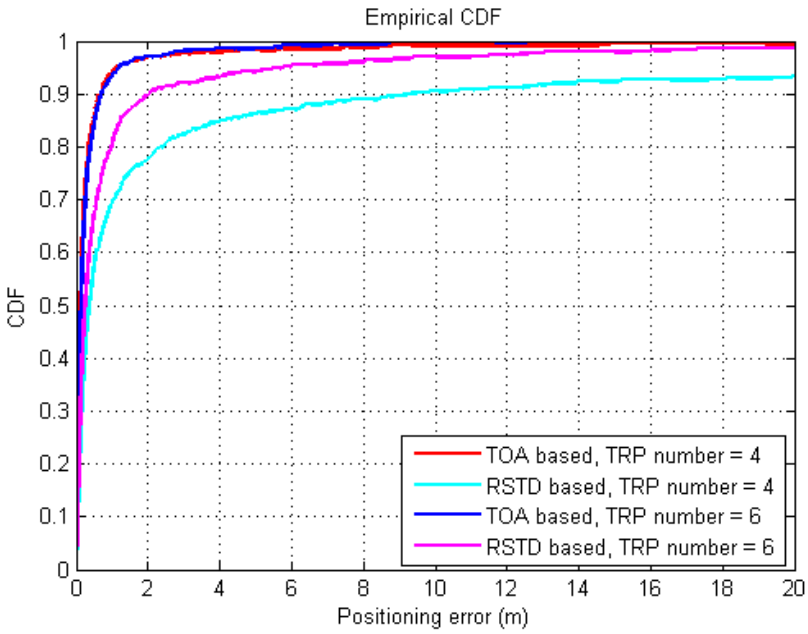


Figure 14 ToA vs. RSTD without UE timing error

Table 10 Evaluation results for TOA based positioning and RSTD based positioning, UE distribution area = [120x60 m]

Measurement for location calculation	**UE timing error**	**Model input**	**Model output**	**Number of TRP for location calculation**	**Positioning accuracy(m) @90% CDF=90%**
TOA	10ns	CIR	TOA	6	1.51
RSTD	10ns	CIR	TOA	6	4.04
TOA	50ns	CIR	TOA	6	10.18
RSTD	50ns	CIR	TOA	6	>15

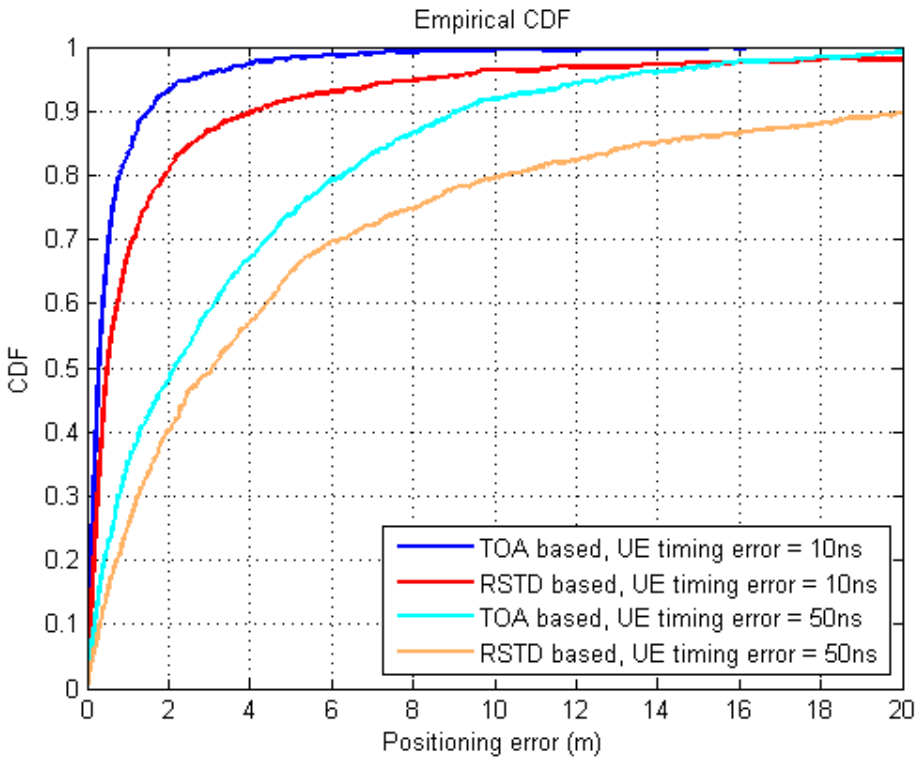


Figure 15 ToA vs. RSTD with UE timing error

Observation 5: ToA as an intermediate report is better than RSTD for AI/ML assisted positioning.

Proposal 3: Support TOA as an intermediate measurement for reporting from UE side to LMF side or from gNB side to LMF side directly.

3.2.3. AI/ML based LOS/NLOS identification

Apart from the mentioned AI/ML based TOA estimation method (CIR-TOA-position), AI/ML based LOS/NLOS identification is another popular positioning scheme with the advantages of great comparability with legacy protocols. In such case, AI/ML technology can be regarded as an enhancement to conventional non-AI LOS/NLOS methods since AI/ML can achieve a more accurate LOS/NLOS identification attached with a confidence metric. Importantly, there is no obvious performance degradation when the AI/ML model associated with a specific TRP is transferred to another TRP, and thus it also enjoys great generalization capability across TRPs. However, compared to the AI/ML based TOA estimation in which AI/ML model is used to estimate TOA directly, its performance still relies on the existence of LOS paths between UE and TRPs for AI/ML based LOS/NLOS identification and may be out of work in heavy NLOS scenarios. Moreover, how to obtain LOS/NLOS labels is a very challenging task for data collection.

Considering limited LOS paths in InF-DH scenarios with clutter parameter {0.6, 6, 2}, we evaluate the positioning performance of AI/ML based LOS/NLOS identification positioning method in InF-DH scenarios with clutter parameter {0.4, 2, 2} where about half of channels are with LOS path. The specific simulation method can refer to the procedure in Figure 17. As shown in Table 11, it is observed that the AI/ML model with CIR as input can achieve more accurate LOS/NLOS identification with comparison to the legacy R17 method, since more potential features of CIR are captured to establish a connection with LOS/NLOS characteristic, such as delay spread (a channel with LOS path usually has smaller delay spread). As shown in Table 12, compared to AI/ML based LOS/NLOS identification, AI/ML based TOA estimation method still has significant performance gain thanks to the powerful capability of AI/ML in TOA feature extraction.

Table 11 Evaluation results of LOS/NLOS identification accuracy for AI/ML model deployed on UE or Network side, without model generalization, full-connection network , UE distribution area = [120x60 m]

Model input	Model output	Label	Clutter param	Dataset size & type	AI/ML complexity	Accuracy of LOS/NLOS identification

				Training	test	Model complexity	Computational complexity	AI/ML
CIR	LOS/NLOS	0	{0.4, 2, 2}	25k	1k	3.62M*18	7.24M*18	>99%
R17 [9]			{0.4, 2, 2}	/				93%

Table 12 Evaluation results for AI/ML model deployed on UE or Network side, without model generalization, full-connection network, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Clutter param**	**Dataset size & type**		**AI/ML complexity**		**Horizontal positioning accuracy at CDF=90% (meters)**
				Training	test	Model complexity	Computational complexity	AI/ML
CIR	LOS/NLOS	0	{0.4, 2, 2}	25k	1k	3.62M*18	7.24M*18	1.10
CIR	TOA	0	{0.4, 2, 2}	25k	1k	44M*18	1.45G*18	0.39

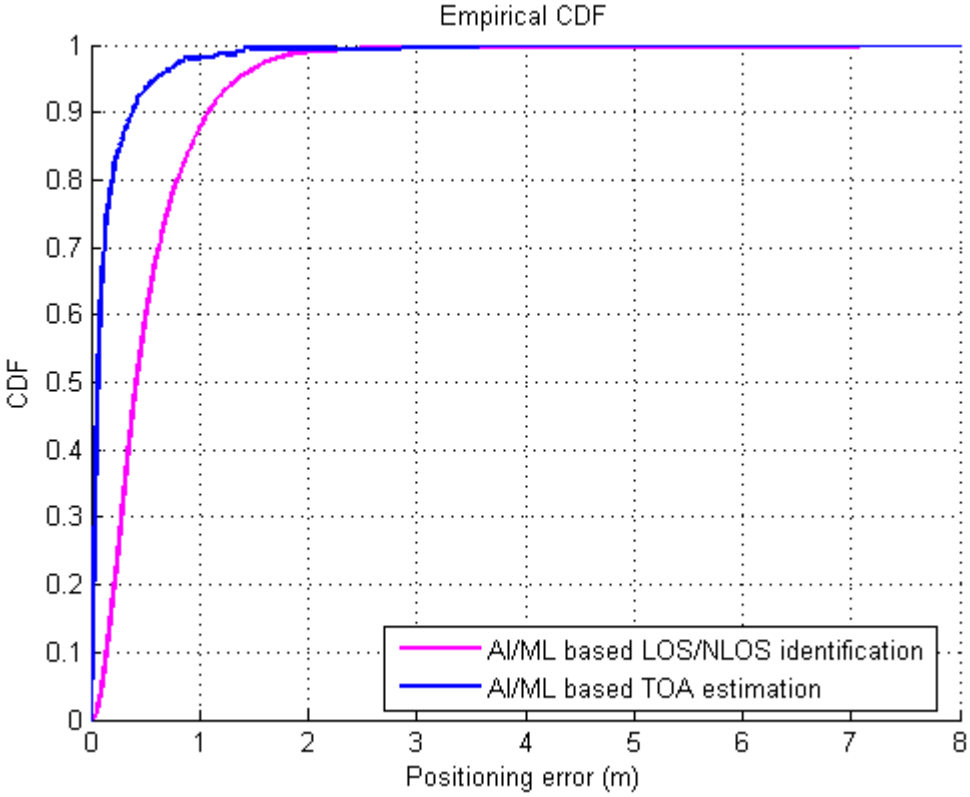


Figure 16 CDF of positioning accuracy of different AI/ML assisted positioning methods

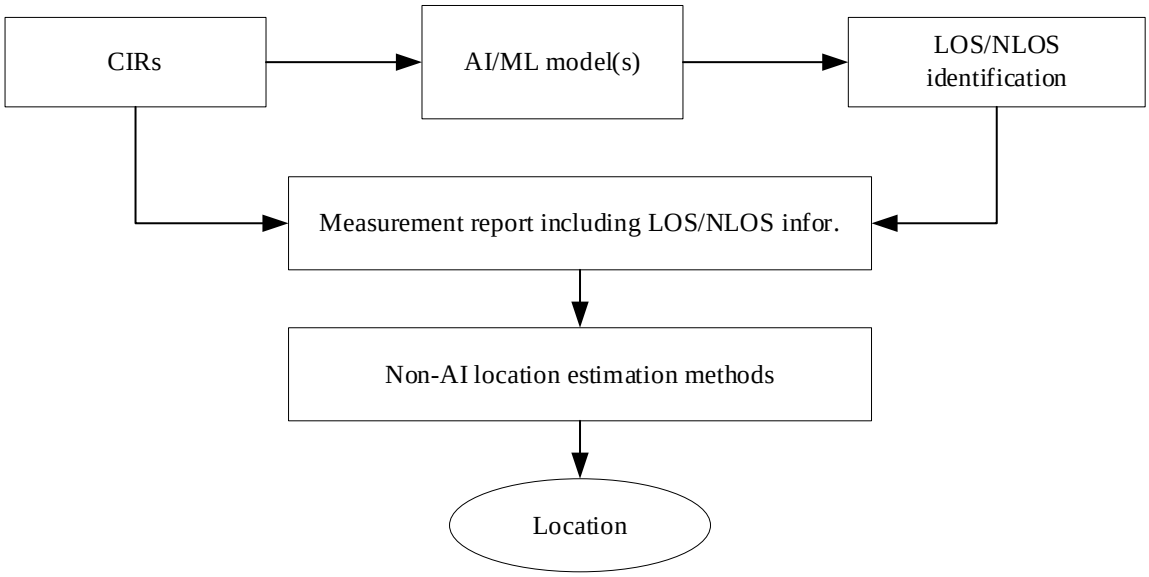


Figure 17 AI/ML based LOS/NLOS identification for positioning

Observation 6: AI/ML based LOS/NLOS identification for positioning has the following advantages:

- **More accurate LOS/NLOS identification along with a confidence metric**
- **Better compatibility with existing positioning protocol framework.**
- **Great generalization capability.**

and disadvantages:

- **Positioning performance could suffer from severe degradation in heavy-NLOS scenarios.**
- **Obtain LOS/NLOS labels is a challenging task for data collection.**

Proposal 4: Capture in the TR the benefits of AI/ML assisted positioning in terms of positioning accuracy and model generalization.

4. Generalization performance evaluation

As we discussed in section 2, AI/ML model generalization performance is greatly important for actual model deployment. In this section, model generalization is evaluated when considering varying settings/scenarios and implementation imperfections.

At the RAN1#110 meeting, it was agreed that:

Agreement

To investigate the model generalization capability, at least the following aspect(s) are considered for the evaluation for AI/ML based positioning:

a) Different drops

- Training dataset from drops {A0, A1,…, AN-1}, test dataset from unseen drop(s) (i.e., different drop(s) than any in {A0, A1,…, AN-1}). Here N>=1.

b) Clutter parameters, e.g., training dataset from one clutter parameter (e.g., {40%, 2m, 2m}), test dataset from a different clutter parameter (e.g., {60%, 6m, 2m});

c) Network synchronization error, e.g., training dataset without network synchronization error, test dataset with network synchronization error;

Other aspects are not excluded.

Note: It's up to participating companies to decide whether to evaluate one aspect at a time, or evaluate multiple aspects at the same time.

In RAN1#110b-e meeting, it was agreed that:

Agreement

To investigate the model generalization capability, the following aspect is also considered for the evaluation of AI/ML based positioning:

(e) InF scenarios, e.g., training dataset from one InF scenario (e.g., InF-DH), test dataset from a different InF scenario (e.g., InF-HH)

Following the above agreements, we further conducted performance evaluations under different settings/scenarios to show their impact to AI/ML model performance.

4.1. Generalization performance for direct AI/ML positioning

4.1.1. Different drops in the same scenario

We perform some simulations to evaluate the generalization capability of direct AI/ML positioning with multi-TRPs' CIRs as input. As shown in Table 13, while the AI/ML model trained with dataset of drop 1 performs well with test dataset of drop 1, the performance would deteriorate severely when the model (without any modification on parameters) is tested on dataset of other drops. It is indicated that AI/ML model suffers from poor generalization capability across different drops for direct AI/ML positioning. Here, the concept 'different *drops*' means different distributions of large-scale parameters in system level simulation, and these large-scale parameters contain absolute time of arrival, angle of arrival, angle of departure, power of LOS/NLOS paths, initial phase of LOS/NLOS paths, delay of LOS/NLOS paths, and so on. For the case of InF scenario, different drops can be intuitively viewed as different factories with different interiors in general.

Table 13 Evaluation results of different drops for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	Drop1	Drop1	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	Drop1	Drop2	25k	1k	1.65M	22.30M	6.00
CIR	Pos.	0	Drop1	Drop3	25k	1k	1.65M	22.30M	5.81

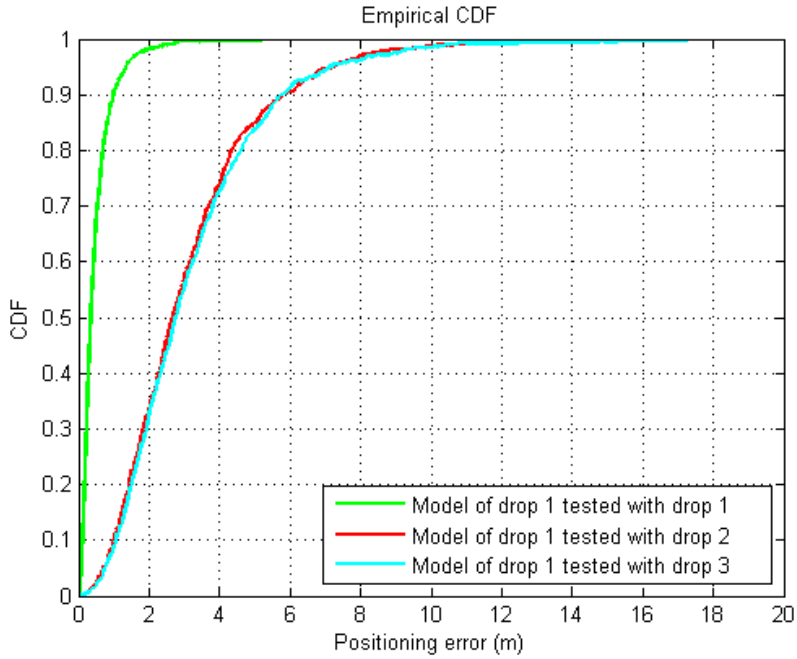


Figure 18 CDF of positioning accuracy when AI model is tested on other drops

Observation 7: Positioning performance of direct AI/ML positioning degrades when the model trained with dataset of one drop is tested with dataset of other drops.

4.1.2. Different clutter parameters

We further evaluate the model generalization performance under clutter parameters {0.6, 6, 2} and {0.4, 2, 2}. As shown in Table 14, AI/ML model performs well when the training dataset and test dataset are generated with the same clutter parameter. However, the positioning performance can drop dramatically when the training dataset and test dataset are generated with different clutter parameters, indicating that AI/ML model suffers from poor generalization capability across different clutter parameters. Moreover, training AI/ML model with a mixed dataset is an effective way to improve generalization performance. It is noted that the mixed dataset has twice the amount of samples as dataset of {0.6, 6, 2} and {0.4, 2, 2}.

Table 14 Evaluation results of different clutter parameters for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	{0.6, 6, 2}	{0.6, 6, 2}	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	{0.6, 6, 2}	{0.4, 2, 2}	25k	1k	1.65M	22.30M	8.67
CIR	Pos.	0	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	1.65M	22.30M	1.06
CIR	Pos.	0	{0.4, 2, 2}	{0.6, 6, 2}	25k	1k	1.65M	22.30M	4.77
CIR	Pos.	0	Mix of {0.6, 6, 2} and {0.4, 2, 2}	{0.6, 6, 2}	25k & 25k	1k	1.65M	22.30M	0.87

CIR	Pos.	0	Mix of {0.6, 6, 2} and {0.4, 2, 2}	{0.4, 2, 2}	25k & 25k	1k	1.65M	22.30M	0.94

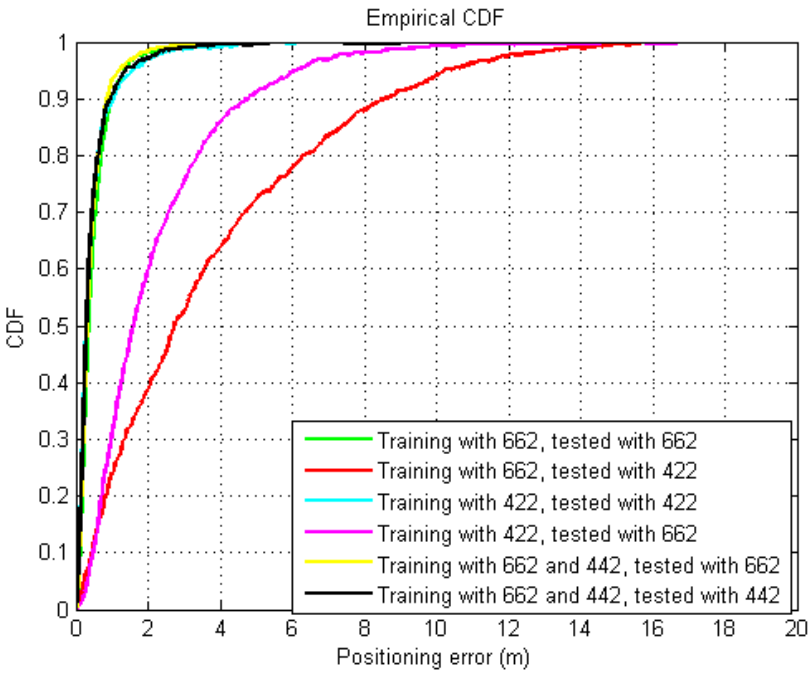


Figure 19 CDF of positioning accuracy of clutter parameters {0.6, 6, 2} and {0.4, 4, 2}

Observation 8: Positioning performance of direct AI/ML positioning degrades when the training and testing datasets are of different clutter parameters in an InF-DH scenario.

Observation 9: Training AI/ML model with a mixed dataset is an effective way to improve model generalization performance.

Proposal 5: Capture in the TR the benefits of training dataset with mixed/different configurations for AI/ML based positioning in terms of AI model generalization capability.

4.1.3. Different scenarios

At the RAN1 #110b-e meeting, it was also agreed that:

Agreement

For AI/ML based positioning, if an InF scenario different from InF-DH is evaluated for the model generalization capability, the selected parameters (e.g., clutter parameters) are compliant with TR 38.901 Table 7.2-4 (Evaluation parameters for InF).

- Note: In TR 38.857 Table 6.1-1 (Parameters common to InF scenarios), InF-SH scenario uses the clutter parameter {20%, 2m, 10m} which is compliant with TR 38.901.

We further evaluate the generalization capability of AI/ML model across different scenarios. From simulation results listed in Table 15, we can observe that AI technology can achieve high-accuracy positioning when training dataset and test dataset are consistent (generated in the same scenario). When the model trained with dataset of an InF-DH scenario is directly transferred to other scenarios, such as InF-HH (100% LOS) and InF-SH scenarios with clutter parameter {20%, 2m, 10m}, the difference between the distributions of training dataset and test dataset will severely deteriorate the positioning accuracy, which indicates that the generalization ability of AI/ML model across scenarios is very limited for direct AI/ML positioning. As we can see, high positioning accuracy (<1m @90%) could be achieved when training dataset and test dataset are sampled from the same scenario. Otherwise, the performance will deteriorate severely (>10m @90%).

Table 15 Evaluation results of different scenarios for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)		Dataset size		AI/ML complexity		Horizontal pos. accuracy at CDF=90% (m)
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	DH	DH	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	HH	HH	25k	1k	1.65M	22.30M	0.63
CIR	Pos.	0	SH	SH	25k	1k	1.65M	22.30M	0.87
CIR	Pos.	0	DH	HH	25k	1k	1.65M	22.30M	>10
CIR	Pos.	0	DH	SH	25k	1k	1.65M	22.30M	>10

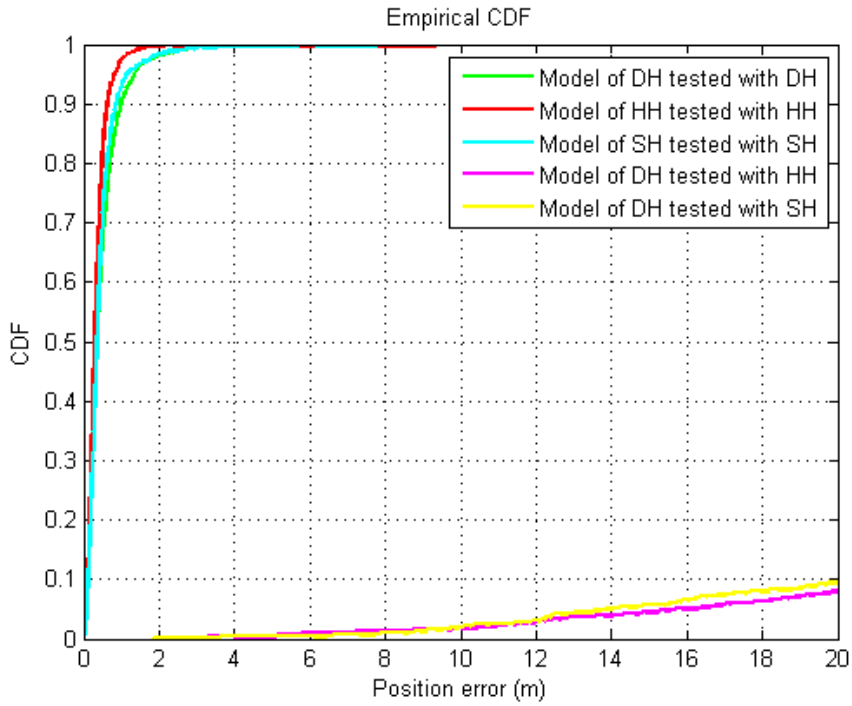


Figure 20 CDF of positioning accuracy when training dataset and test dataset are not matched

Observation 10: The positioning accuracy of direct AI/ML positioning trained with dataset from one InF scenario is seriously degraded when tested on dataset from a different InF scenario.

4.2. Generalization performance for AI/ML assisted positioning

At the RAN1#111 meeting, it was agreed that:

Agreement

- For AI/ML assisted approach, for a given AI/ML model design (e.g., input, output, single-TRP vs multi-TRP), identify the generalization aspects where model fine-tuning/mixed training dataset/model switching is necessary.

4.2.1. Different drops in the same scenario

We evaluate the generalization capability of AI/ML model across drops for AI/ML assisted positioning. As shown in Table 16, it is observed that while the AI/ML model trained with dataset of drop 1 performs well with test dataset of drop 1, the performance will deteriorate severely when the model (without any modification on parameters) is tested on dataset of other drops. This is because AI/ML assisted positioning still relies on fingerprint features of CIR in heavy-NLOS scenarios. Once spatial consistency changes, fingerprint features learned from original training dataset would not adapt to the new test dataset.

Table 16 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Drop1	Drop1	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	Drop1	Drop2	25k	1k	4.26M*18	8.50M*18	10.37

Table 17 CDF of estimation accuracy of intermediate feature TOA (meter)

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Drop1	Drop1	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	Drop1	Drop2	25k	1k	4.26M*18	8.50M*18	8.00

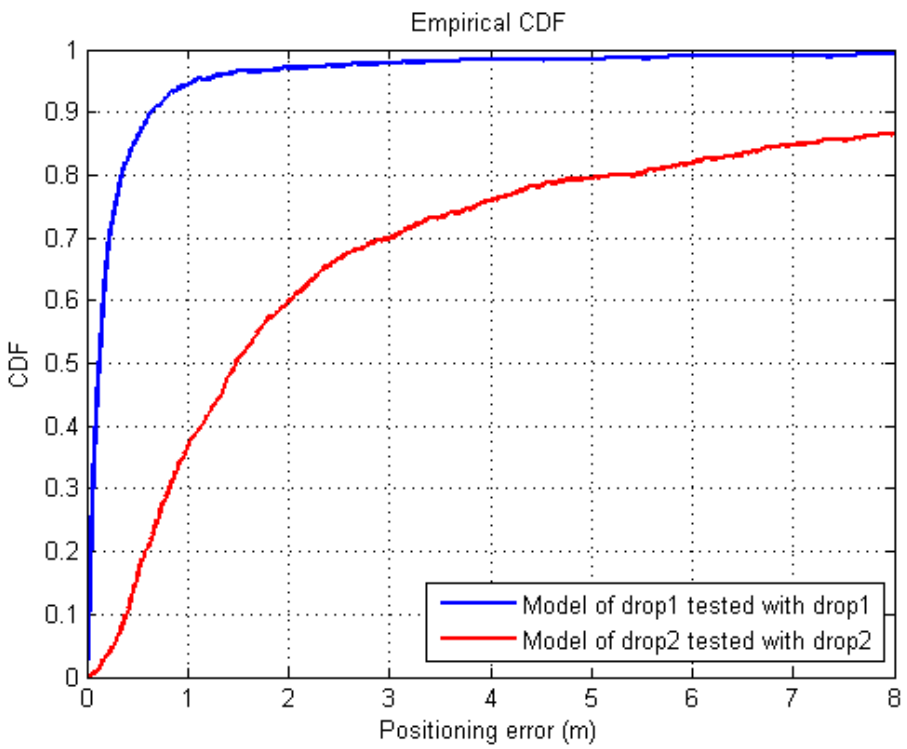


Figure 21 CDF of positioning accuracy when AI model is tested on other drops

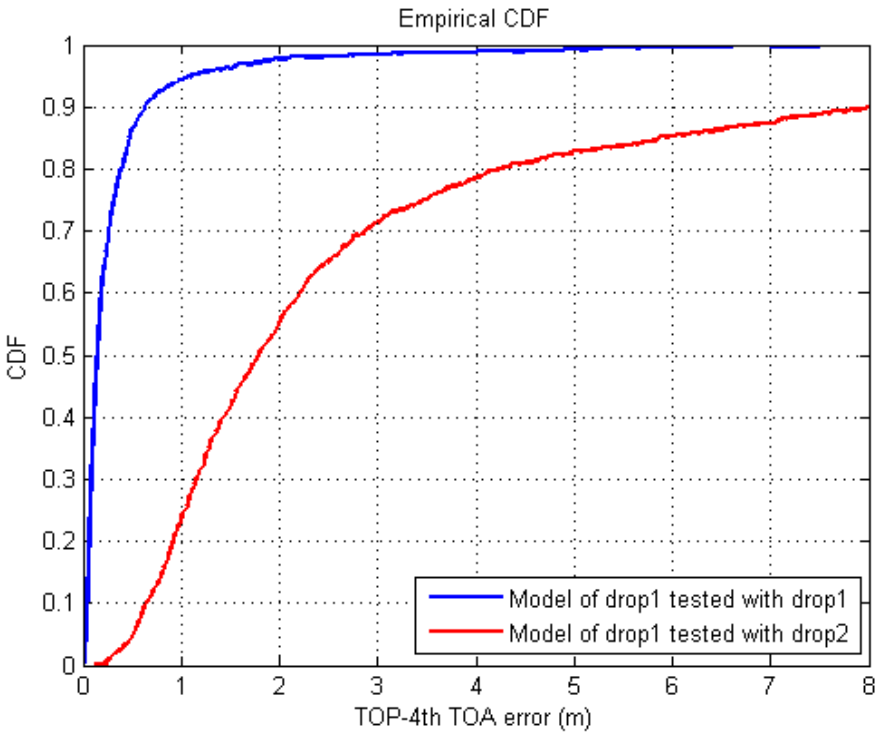


Figure 22 CDF of TOA estimation accuracy when AI model is tested on other drops

Observation 11: Positioning performance of AI/ML assisted positioning degrades when the model trained with dataset of one drop is tested with dataset of other drops.

4.2.2. Different clutter parameters

We evaluate the generalization capability of AI/ML model across clutter parameters for AI/ML assisted positioning. As shown in Table 18, it is observed that while AI/ML model performs well when training dataset and test dataset are sampled from the same clutter parameter configuration. Moreover, the performance would be degraded when the model (without any modification on parameters) is tested with the dataset from a different clutter parameter but still greatly better than that of direct AI/ML positioning as presented in Table 14. This is because the fingerprint features learned from DH{0.6, 6, 2} is still valid for partial NLOS TRPs in DH{0.4, 2, 2}, and utilizing the estimated TOAs related to these TRPs to calculate the final position can reap higher positioning accuracy than pure-fingerprint direct AI/ML positioning. Thus, AI/ML based TOA estimation enjoys better generalization capability as compared to direct AI/ML positioning across clutter parameters.

Table 18 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	{0.6, 6, 2}	{0.6, 6, 2}	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	{0.6, 6, 2}	{0.4, 2, 2}	25k	1k	4.26M*18	8.50M*18	3.70
CIR	TOA	0	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	4.26M*18	8.50M*18	0.32
CIR	TOA	0	{0.4, 2, 2}	{0.6, 6, 2}	25k	1k	4.26M*18	8.50M*18	1.53

Table 19 CDF of estimation accuracy of intermediate feature TOA (meter)

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)		Dataset size		AI/ML complexity		TOP-4th TOA accuracy (m)
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	{0.6, 6, 2}	{0.6, 6, 2}	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	{0.6, 6, 2}	{0.4, 2, 2}	25k	1k	4.26M*18	8.50M*18	6.00
CIR	TOA	0	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	4.26M*18	8.50M*18	0.39
CIR	TOA	0	{0.4, 2, 2}	{0.6, 6, 2}	25k	1k	4.26M*18	8.50M*18	2.30

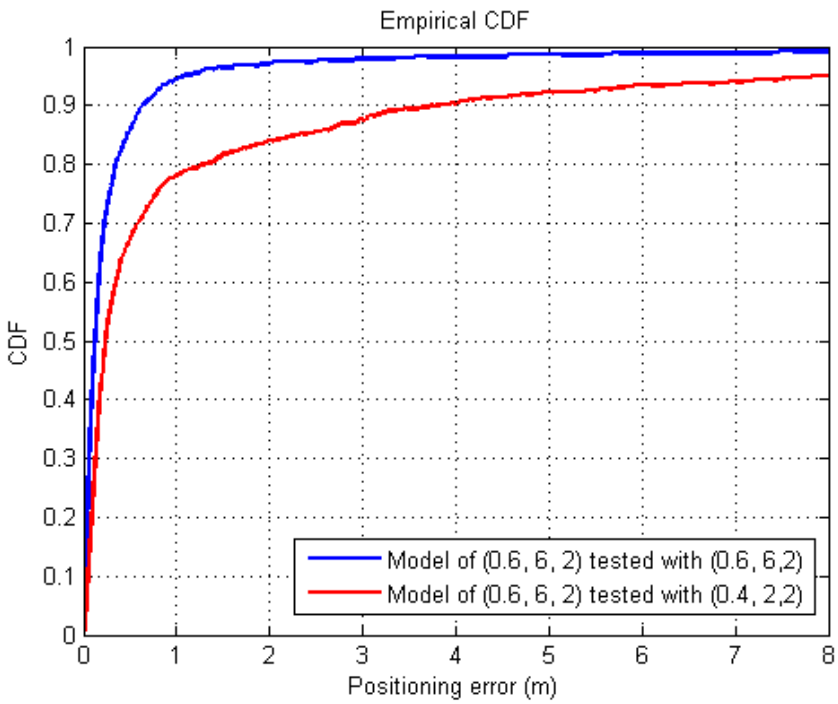


Figure 23 CDF of positioning accuracy when AI model is tested on other clutter

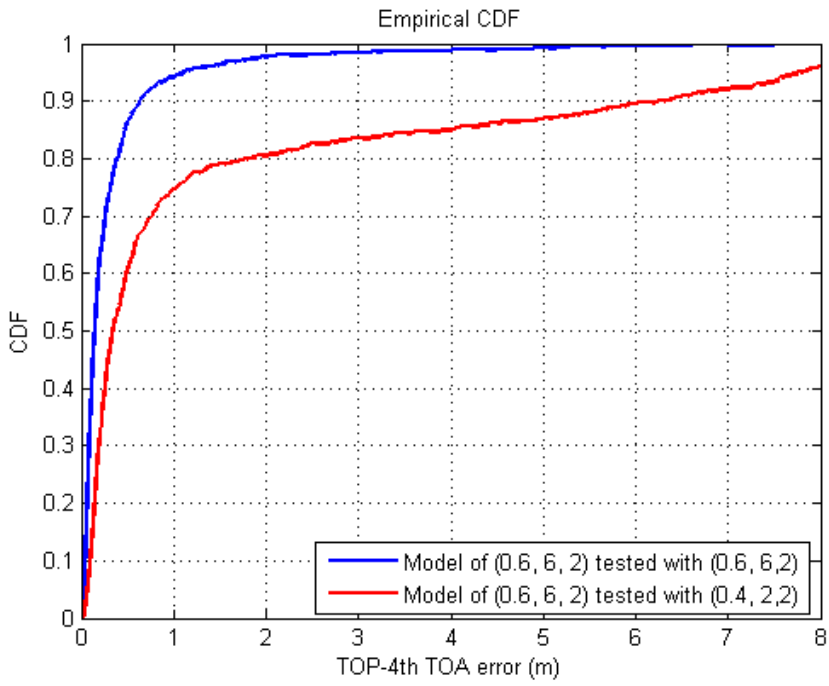


Figure 24 CDF of TOA estimation accuracy when AI model is tested on other clutter

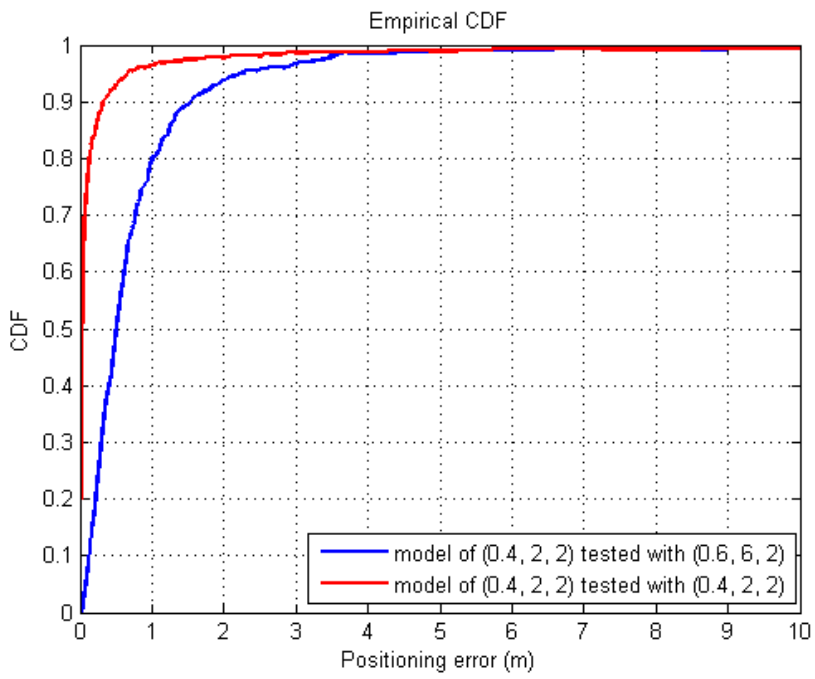


Figure 25 CDF of positioning accuracy when AI model is tested on other clutter

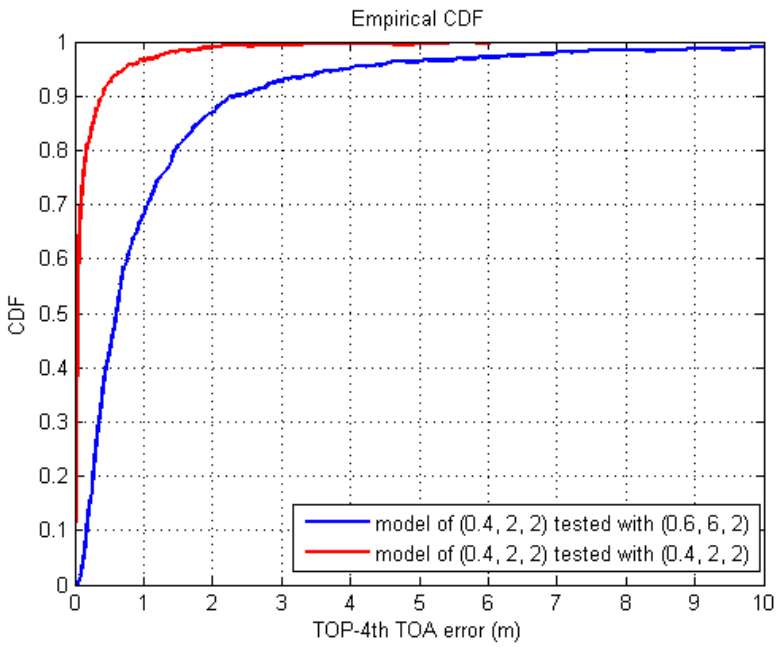


Figure 26 CDF of TOA estimation accuracy when AI model is tested on other clutter

Observation 12: Positioning performance of AI/ML assisted positioning is slightly degraded but still acceptable when the model trained with dataset of one clutter parameter is tested with dataset of another clutter parameter.

Observation 13: AI/ML assisted positioning enjoys better generalization performance than direct AI/ML positioning across clutter parameters.

4.2.3. Different scenarios

We evaluate the generalization capability across scenarios for AI/ML assisted positioning. As shown in Table 20, it is observed that while the AI/ML model trained with dataset of DH performs well with test dataset of DH, the performance will deteriorate severely when the model (without any modification on parameters) is tested on dataset from other scenarios, e.g., HH and SH. This is because the change of TRPs' distribution destroys fingerprint features learned from DH{0.6, 6, 2}. It is worth noting that AI/ML model of HH still reaps very high positioning accuracy when tested in the SH scenario, since AI/ML model estimates TOA based on the first-path delay of CIR mainly not just the fingerprint features. In this way, AI/ML model can achieve accurate TOA estimation for these TRPs with LOS paths, and the final location can be estimated using these accurate TOA estimations. Therefore, at least for those scenarios whose positioning does not rely on fingerprint features, AI/ML based TOA estimation has better generalization ability than direct AI/ML positioning.

Table 20 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	DH	DH	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	DH	HH	25k	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0	DH	SH	25k	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0	HH	SH	25k	1k	4.26M*18	8.50M*18	0.05

Table 21 CDF of estimation accuracy of intermediate feature TOA (meter) , UE distribution area = [120x60 m]

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)		Dataset size		AI/ML complexity		TOP-4th TOA accuracy (m)
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	DH	DH	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	DH	HH	25k	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0	DH	SH	25k	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0	HH	SH	25k	1k	4.26M*18	8.50M*18	0.04

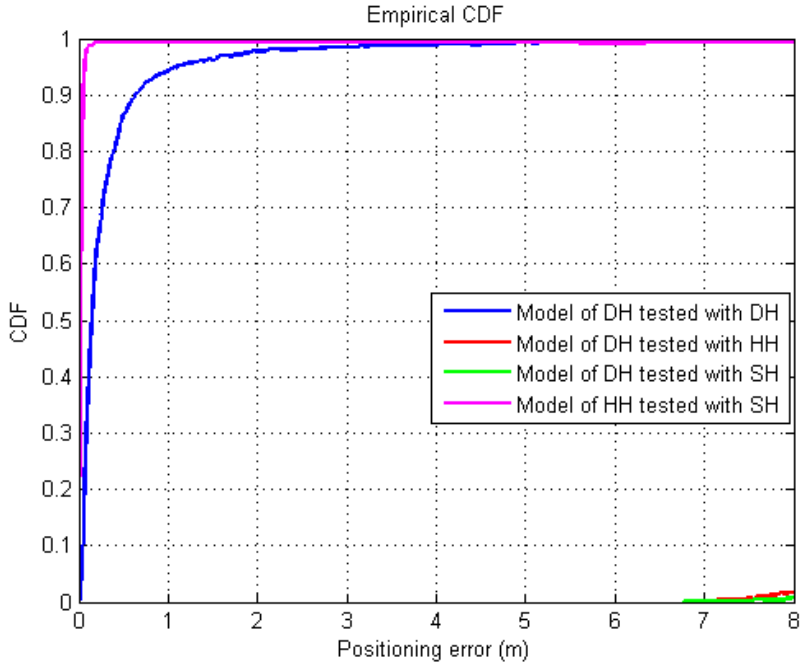


Figure 27 CDF of positioning accuracy when AI model is tested on other scenarios

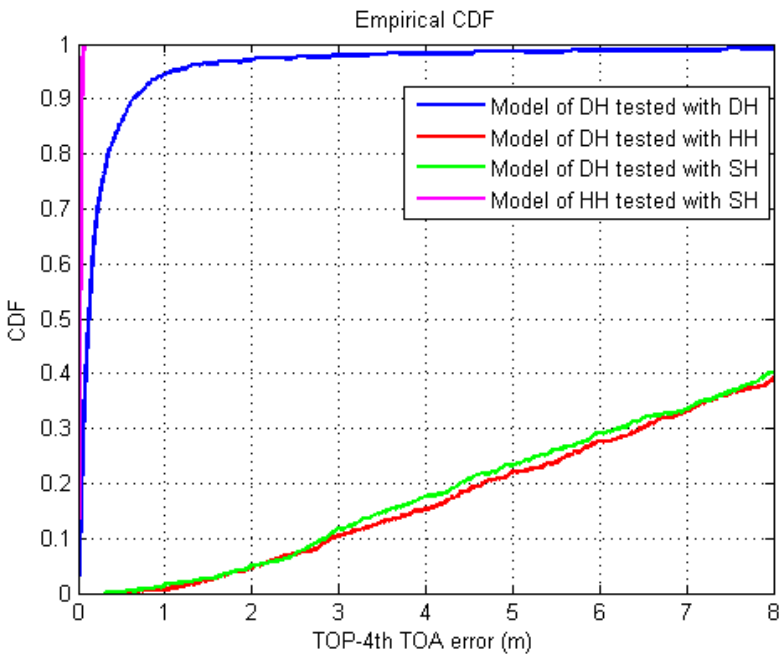


Figure 28 CDF of TOA estimation accuracy when AI model is tested on other scenarios

Observation 14: Positioning performance of AI/ML assisted positioning is degraded when the model trained with dataset of DH is tested with datasets of SH and HH.

Observation 15: For those scenarios whose positioning does not rely on fingerprint features, AI/ML based TOA estimation has better generalization ability than direct AI/ML positioning.

Observation 16: AI/ML based TOA estimation has great advantages in positioning performance, deployment flexibility, compatibility with existing positioning protocol framework, and generalization capability.

4.3. The impact of implementation imperfections for direct AI/ML positioning

At the RAN1#110 meeting, it was agreed that：

Agreement

For AI/ML-based positioning, study impact from implementation imperfections.

At the RAN1#111 meeting, it was agreed that：

Agreement

For AI/ML based positioning, company optionally evaluate the impact of at least the following issues related to measurements on the positioning accuracy of the AI/ML model. The simulation assumptions reflecting these issues are up to companies.

- SNR mismatch (i.e., SNR when training data are collected is different from SNR when model inference is performed).
- Time varying changes (e.g., mobility of clutter objects in the environment)
- Channel estimation error

In section 4.1, we have evaluated the generalization capability of AI/ML model from a high-level perspective, including the generalization of AI/ML model in different drops, different clutters and different scenarios. In practice, other factors stemming from implementation imperfections, such as CIR estimation error, synchronization error and labeling error, can also impair the positioning accuracy even if the deployed AI/ML model is well-trained offline in advance. Indeed, these imperfect factors are unavoidable and difficult to eliminate by regular manners. In this section, in order to assess the unknown risks from the perspective of implementations, we specifically evaluate the impact of these implementation imperfections on positioning performance for direct AI/ML positioning, and propose a potential solution to mitigate these impacts as much as possible.

4.3.1. CIR estimation error

At the RAN1 #110b-e meeting, we have reached the following conclusion:

Conclusion

For evaluation of AI/ML based positioning, it’s up to each company to take into account the channel estimation error in their evaluation. Companies describe the details of their simulation assumption, e.g., realistic or ideal channel estimation, error models, receiver algorithms.

It has been observed that adopting CIR as the input to AI/ML model reaps the best inference accuracy for both direct AI/ML positioning and AI/ML assisted positioning frameworks due to the rich information contained, such as first-path feature and fingerprint feature. The existing schemes are all evaluated under the assumption that ideal CIRs used for model training and inference can be obtained while ignoring the implementation imperfections. In practice, CIR estimation error is always existed and it is impossible to obtain the ideal CIR by RF measurement. Here, we focus on the evaluation of impact of CIR estimation error on positioning performance for direct AI/ML positioning.

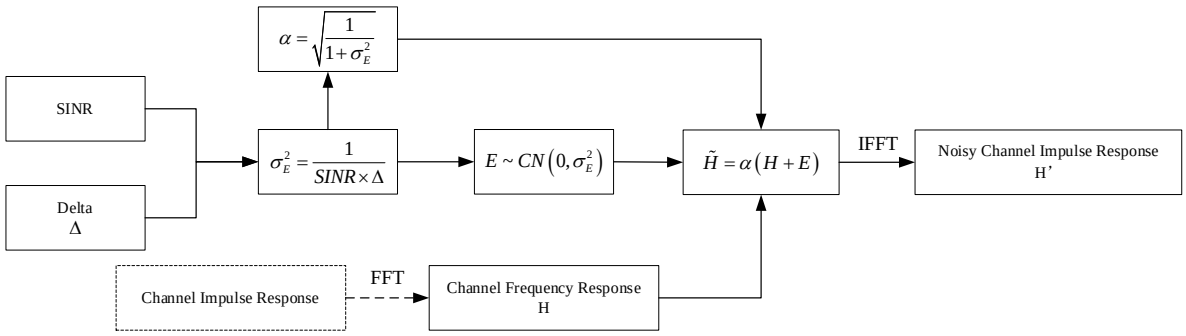


Figure 29 A procedure of modeling CIR estimation error [10]

The performance of channel estimation is mainly affected by interference and noise. As shown in Figure 29, we use a procedure of adding channel estimation error to time-domain CIR with reference to [10], in which the additional estimation error obeying a zero-mean Complex Gaussian distribution is generated according to the received SINR. Without loss of generality, the compensation factor Δ is set to 9dB in our simulation setting. To estimate the dynamic range of SINR for the considered InF-DH scenario, we further calculate the distributions of SINR when assuming there exists different number of interfering TRPs:

- Without interference, i.e., all TRPs will not interference with each other:

$$\mathrm{SINR}_n = 10\log_{10}\frac{RSRP_n}{\mathrm{noise}}$$

- With N interfering TRPs:

$$\mathrm{SINR}_n = 10\log_{10}\frac{RSRP_n}{\sum_{i\in I} RSRP_i + \mathrm{noise}}$$

where $RSRP_i$ denotes the RSRP between UE and i-th TRP, and I is a set containing all interfering TRPs (excluding the target TRP_n). The noise is calculated as follows:

$$noise(dBm) = -174dBm + 10\lg Bandwidth + 7dB$$

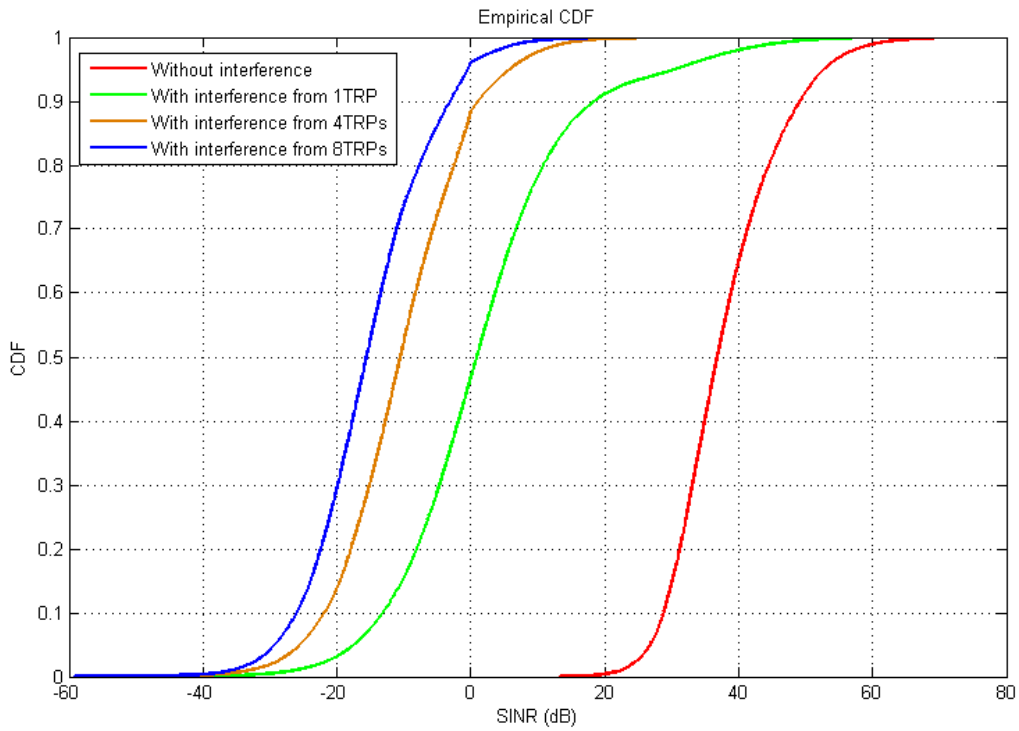


Figure 30 Dynamic range of SINR for the InF-DH scenario

The distributions of SINR with 0, 1, 4, 8 interfering TRPs are shown in Figure 30. By the way, the interfering TRPs are selected randomly from other 17 TRPs. Clearly, when there is no interference, SINR is ranging from 20dB to 60dB. However, SINR will dramatically decrease even when there is only one interfering TRP since interference dominates SINR as compared to noise. Considering that poor SINR condition can severely deteriorate channel estimation quality and cause unknown channel estimation error, we strongly believe evaluating the impact of CIR estimation error on positioning performance is very necessary at least when CIR or PDP is adopted as model input.

In practice, the SINR condition of training dataset and test dataset may not remain the same due to the dynamic wireless environment. In this regard, we further evaluate the positioning performance when training dataset and test dataset are sampled from different SINR conditions. Considering that some dedicated reference signals can be configured for high-quality data collection, we assume that training dataset comes from a high-SINR condition without interference and test datasets suffer from the interference from various number of TRPs. As shown in Table 22, it is observed that the impact of noise is negligible while the interference from other TRPs can severely deteriorate the positioning accuracy. The reason behind is that the additional channel estimation error caused by interference impairs the spatial consistency, making partial mismatch between training dataset and test dataset, while fingerprint feature is of great importance for direct AI/ML positioning.

Table 22 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	Without interference	0 interfering TRP (Without interference)	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0		1 interfering TRP	25k	1k	1.65M	22.30M	8.35
CIR	Pos.	0		4 interfering TRPs	25k	1k	1.65M	22.30M	10.22
CIR	Pos.	0		8 interfering TRPs	25k	1k	1.65M	22.30M	13. 14

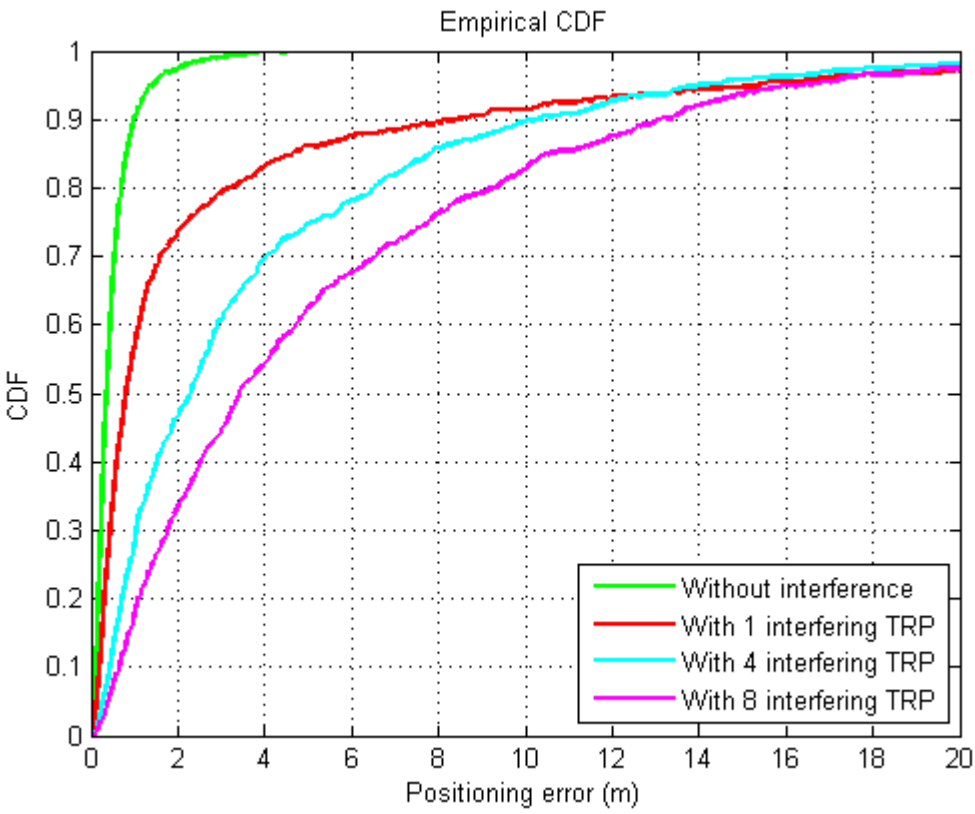


Figure 31 Evaluation of the impact of CIR estimation error on positioning accuracy

Observation 17: The interference from TPRs can dramatically impair the positioning performance of AI/ML model.

Proposal 6: Further study the impact and potential solution of CIR estimation error on AI/ML based positioning performance.

4.3.2. Synchronization error

At the RAN1#110 meeting, it was agreed that:

Agreement

To investigate the model generalization capability, at least the following aspect(s) are considered for the evaluation for AI/ML based positioning:

- Network synchronization error, e.g., training dataset without network synchronization error, test dataset with network synchronization error;

Synchronization error caused by hardware imperfection is another imperfect factor affecting the generalization performance of AI/ML model. As we analyzed earlier, AI/ML model performs positioning inference with reference to three features of CIR, including first-path information due to the existence of absolute time of arrival, fingerprint information due to the existence of spatial consistency, and correlation of CIRs for fixed TRPs' topology. Intuitively, synchronization error can directly impair the feature of first-path delay. Then, it can partally impair the spatial consistency, resulting in the dissimilarity of CIRs for users in close proximity to each other. Finally, it can impair the correlation of CIRs for fixed TRPs' topology since synchronization errors may be different across TRPs. Moreover, synchronization error is unavoidable and difficult to eliminate completely, and thus it is necessary to evaluate its impact on positioning performance for AI/ML based positioning.

Assume that training dataset is sampled with perfect synchronization and test dataset is sampled with 2ns, 10ns and 50ns synchronization errors. This assumption is reasonable since synchronization error can be mitigated very well in the process of data collection, such as dedicated RS configuration and data post-processing, but it is difficult to estimate real-time and accurate synchronization error in the deployed scenario.

As shown in Table 23, it is noticeable that synchronization error can dramatically deteriorate the positioning performance of AI/ML model for direct AI/ML positioning. Meanwhile, the positioning accuracy significantly degrades with the increase of synchronization error. Therefore, the impact of synchronization error on positioning performance can not be ignored.

Table 23 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	0ns	0ns	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	0ns	2ns	25k	1k	1.65M	22.30M	1.64
CIR	Pos.	0	0ns	10ns	25k	1k	1.65M	22.30M	4.56
CIR	Pos.	0	0ns	50ns	25k	1k	1.65M	22.30M	10.18

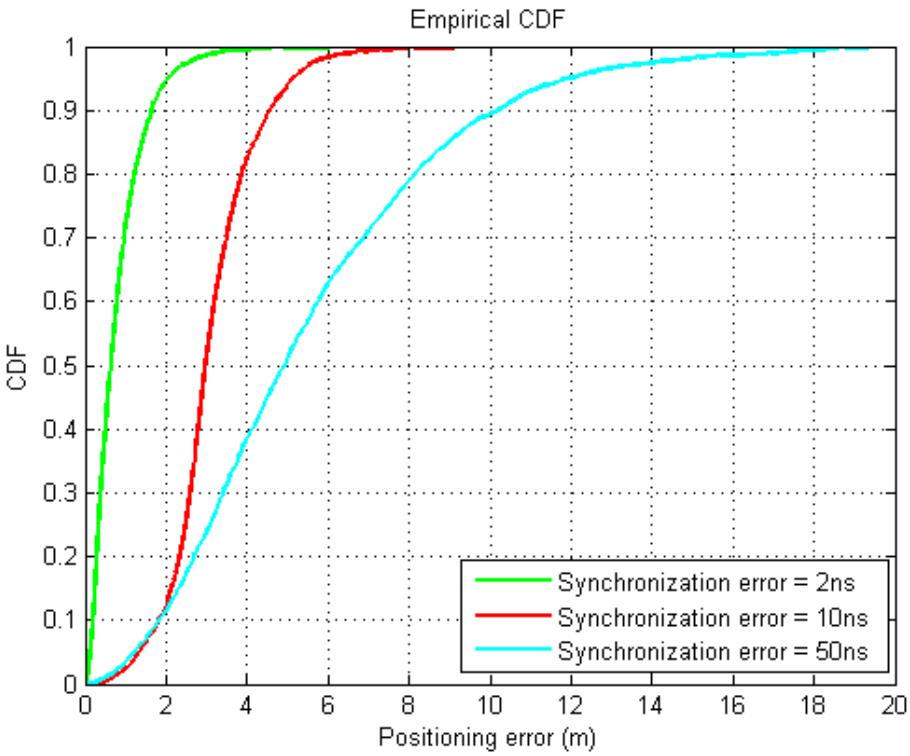


Figure 32 Evaluation of the impact of synchronization error on positioning accuracy

Regarding the serious impairment on positioning performance, it is meaningful to study the solution to mitigate the impact of synchronization error. From the perspective of AI/ML technology, we propose an efficient solution by mix-training. Specifically, in addition to training data with perfect synchronization, some samples with synchronization error are additionally included into the training dataset. These samples with synchronization error can be collected from the real environment or obtained through data augmentation of existing data. As shown in Table 24, when only 2k samples with synchronization error 50ns are added into the training dataset, the positioning accuracy of AI/ML model is significantly improved from 10.18m@90% to 1.52m@90%, proving that mix-training can deal with synchronization error efficiently. The reason is that AI/ML model can learn the difference in training data with various synchronization errors via mix-training.

Table 24 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML

CIR	Pos.	0	0ns	10ns	25k	1k	1.65M	22.30M	4.56
CIR	Pos.	0	Mix 0ns+10ns	10ns	25k+2k	1k	1.65M	22.30M	1.16
CIR	Pos.	0	0ns	50ns	25k	1k	1.65M	22.30M	10.18
CIR	Pos.	0	Mix 0ns+50ns	50ns	25k+2k	1k	1.65M	22.30M	1.52

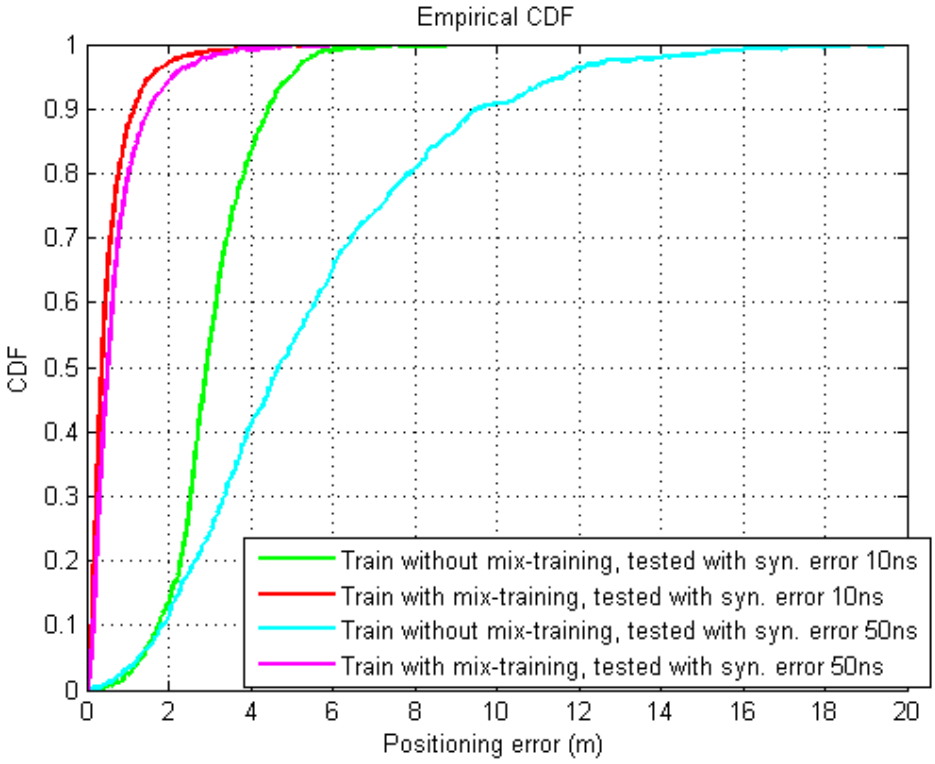


Figure 33 Evaluation of the impact of synchronization error on positioning accuracy

Observation 18: The positioning accuracy of direct AI/ML positioning significantly degrades with the increase of network synchronization error.

Observation 19: The positioning accuracy of AI/ML model is significantly improved from 10.18m@90% to 1.52m@90% by mix-training with samples of synchronization error.

Proposal 7: Further study the impact and potential solution of network synchronization error for the performance of direct AI/ML positioning.

4.3.3. Labeling error

At the RAN1#110 meeting, it was agreed that:

Agreement

When providing evaluation results for AI/ML based positioning, participating companies are expected to describe data labelling details, including:

- Imperfection of the ground truth labels, if any

At the RAN1#112 meeting, it was agreed that:

Agreement

For direct AI/ML positioning, study the impact of labelling error to positioning accuracy

- The ground truth label error in each dimension of x-axis and y-axis can be modeled as a truncated Gaussian distribution with zero mean and standard deviation of L meters, with truncation of the

distribution to the [-2*L, 2*L] range.
    - Value L is up to sources.
- Other models are not precluded
- [Whether/how to study the impact of labelling error to label-based model monitoring methods]
- [Whether/how to study the impact of labelling error for AI/ML assisted positioning.]

Regarding the measurement error, 100 percent correct ground truth label is not always available and labeling error may exist. To some extent, training AI/ML model with these noisy labels may severely impair the positioning performance due to the existence of wrong prior knowledge in training dataset. Therefore, it is meaningful to evaluate the impact of labeling error on positioning performance for AI/ML based positioning.

For the convenience of theoretical analysis, the method of adding labeling error to ground truth label is specified as follows：

$$(x', y') = (x, y) + (e_x, e_y)$$

where (x, y) denotes the coordinate in the horizontal direction, (e_x, e_y) denotes the labeling error obeying Gaussian distribution without truncation, and (x', y') denotes the noisy label with labeling error.

As shown in Table 25, the positioning accuracy gradually degrades with the increase of labeling error, but is still acceptable until standard deviation σ is 1 m (2.17m@90%). It is observed that AI/ML based positioning is robust to label noise to some extent. For example, when the standard deviation σ of labeling error is 4m, the theoretical error of positioning accuracy is about 8.50m@90%, which is larger than that of AI/ML based positioning 5.13m@90%. From this perspective, AI/ML model can also act as a filter, filtering out the noise of training data to find the true pattern partially. Therefore, according to the requirement of positioning accuracy, the maximum acceptable labeling error should be identified firstly before data collection. For example, the maximum acceptable labeling errors (standard deviation) in the horizontal direction should be less than 1m to achieve 2m@90% positioning accuracy.

Table 25 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	Std = 0	0	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	Std = 0.5	0	25k	1k	1.65M	22.30M	1.51
CIR	Pos.	0	Std = 1	0	25k	1k	1.65M	22.30M	2.17
CIR	Pos.	0	Std = 2	0	25k	1k	1.65M	22.30M	3.55

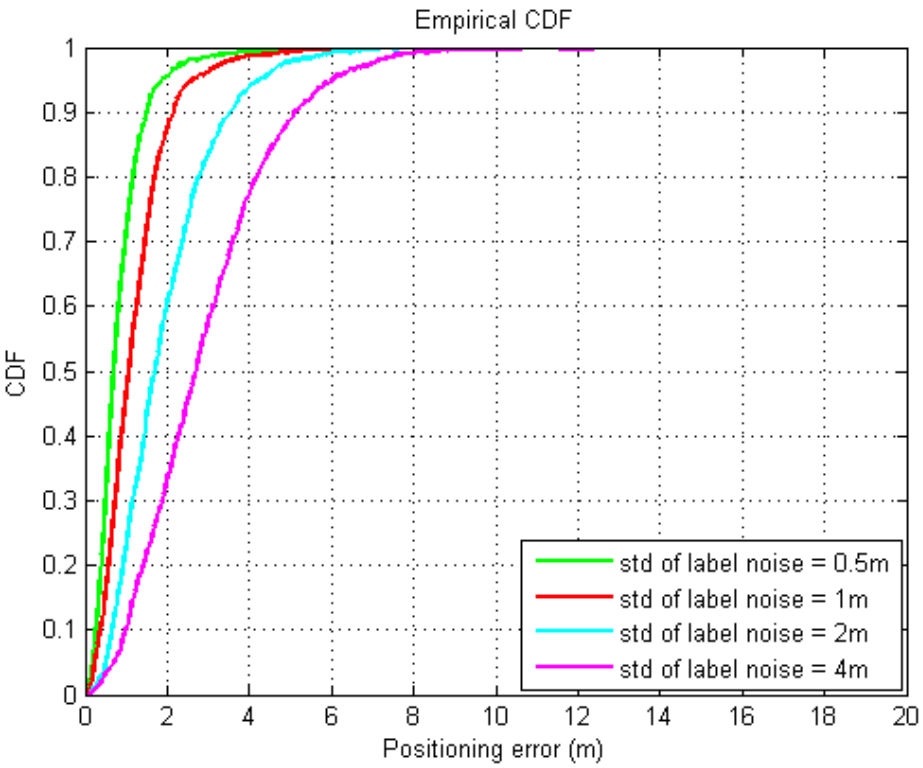


Figure 34 Evaluation of the impact of labeling error on positioning accuracy

Following the agreement on labeling error of RAN #112, we also evaluate the degradation of positioning performance when the labeling error is modeled as truncated Gaussian distribution rather than Gaussian distribution. As shown in Table 26, similar to modeling labeling error as Gaussian distribution, the positioning accuracy gradually degrades with the increase of labeling error, but is still acceptable until standard deviation σ is 1 m (2.20m@90%)

Table 26 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	Std = 0	0	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	Std = 0.5	0	25k	1k	1.65M	22.30M	1.66
CIR	Pos.	0	Std = 1	0	25k	1k	1.65M	22.30M	2.20
CIR	Pos.	0	Std = 2	0	25k	1k	1.65M	22.30M	3.47

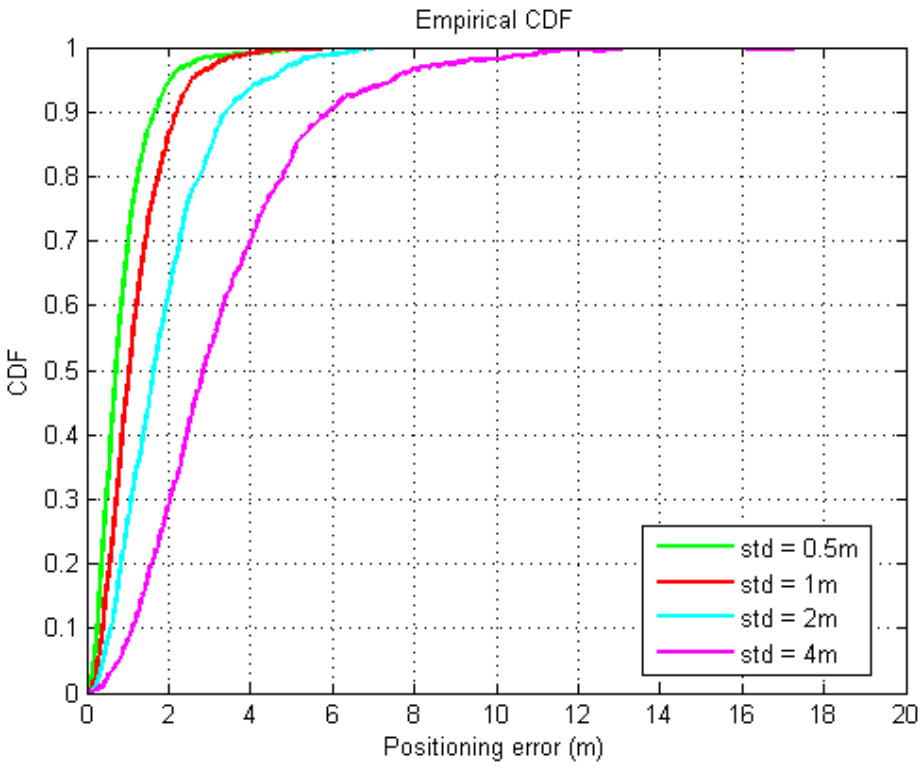


Figure 35 Evaluation of the impact of labeling error on positioning accuracy

Observation 20: The positioning accuracy gradually degrades with the increase of labeling error, but is still acceptable until standard deviation σ is 1 m. The maximum acceptable labeling errors (standard deviation) in the horizontal direction should be less than 1m to achieve 2m@90% positioning accuracy.

Observation 21: AI/ML based positioning is robust to label noise to some extent.

Proposal 8: According to the requirement of positioning accuracy, the maximum acceptable labeling error should be identified firstly before data collection

Proposal 9: Further study the impact and potential solution of labeling error on AI/ML based positioning performance

4.3.4. UE timing error

At the RAN1#110 meeting, it was agreed that:

Offline Agreement

To investigate the model generalization capability, the following aspect is also considered for the evaluation of AI/ML based positioning:

a) UE/gNB RX and TX timing error.

The baseline non-AI/ML method may enable the Rel-17 enhancement features (e.g., UE Rx TEG, UE RxTx TEG).

UE timing error caused by hardware imperfection of UE terminal is another imperfect factor affecting the generalization performance of AI/ML model. As we analyzed earlier, AI/ML model performs positioning inference with reference to three features of CIR, including first-path information due to the existence of absolute time of arrival, fingerprint information due to the existence of spatial consistency, and correlation of CIRs for fixed TRPs' topology. Intuitively, UE timing error can directly impair the feature of first-path delay, making the first-path delay related to all TRPs derivate from the true values as a while. Then, it can partially destroy the spatial consistency, resulting in the dissimilarity of CIRs for users in close proximity to each other. Moreover, UE timing error is unavoidable and difficult to eliminate completely, and thus it is necessary to evaluate its impact on positioning performance for AI/ML based positioning.

Assume that training dataset is sampled with perfect UE timing error and test dataset is sampled with 2ns, 10ns and 50ns timing errors. This assumption is reasonable since UE timing error can be mitigated very well in the process of data collection, such as dedicated RS configuration and data post-processing, but it is difficult to estimate accurate timing error in real-time in the deployed scenario.

As shown in Table 23 and Table 27, compared with the impact of synchronization error, UE timing error can slightly degrade the positioning performance of AI/ML model. The reason behind is that UE timing error does not destroy the relative relationship of CIRs among TRPs, but only makes a common time offset for all CIRs. Fortunately, AI/ML model is robust to this offset due to its translation invariance. However, the positioning accuracy gradually degrades with the increase of UE timing error. Therefore, the impact of UE timing error on positioning performance can not be ignored in practice.

Table 27 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	0ns	0ns	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	0ns	2ns	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	0	0ns	10ns	25k	1k	1.65M	22.30M	1.24
CIR	Pos.	0	0ns	50ns	25k	1k	1.65M	22.30M	3.45

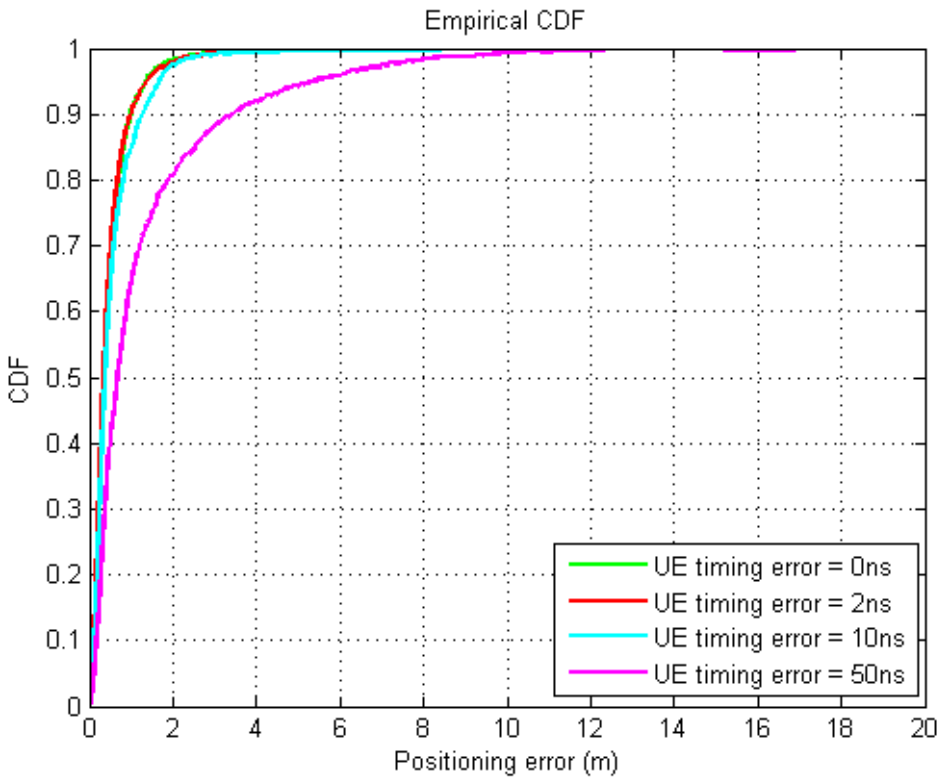


Figure 36 Evaluation of the impact of UE timing error on positioning accuracy

Regarding its impairment on positioning performance, it is meaningful to study the solution to mitigate the impact of UE timing error. From the perspective of AI/ML technology, data augmentation may be a feasible solution. Specifically, in addition to training data with perfect timing, some samples with timing error are additionally included into the training dataset. These samples with timing error can be collected from the real environment or obtained through data augmentation of the existing data. As shown in Table 28, when only 2k samples with UE timing error 50ns are added into the training dataset, the positioning accuracy of AI/ML model is significantly improved from 3.45m@90% to 2.04m@90%, proving that mix-training can deal with UE timing error efficiently. The reason is that AI/ML model can learn the difference in training data with various UE timing errors via mix-training. It is expected that a more robust AI/ML model can be generated via mixing more samples with various UE timing errors into the training dataset.

Table 28 Evaluation results for AI/ML model deployed on UE or Network side, with model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0	0ns	10ns	25k	1k	1.65M	22.30M	1.24
CIR	Pos.	0	Mix 0ns+10ns	10ns	25k+2k	1k	1.65M	22.30M	1.18
CIR	Pos.	0	0ns	50ns	25k+2k	1k	1.65M	22.30M	3.45
CIR	Pos.	0	Mix 0ns+50ns	50ns	25k+2k	1k	1.65M	22.30M	2.04

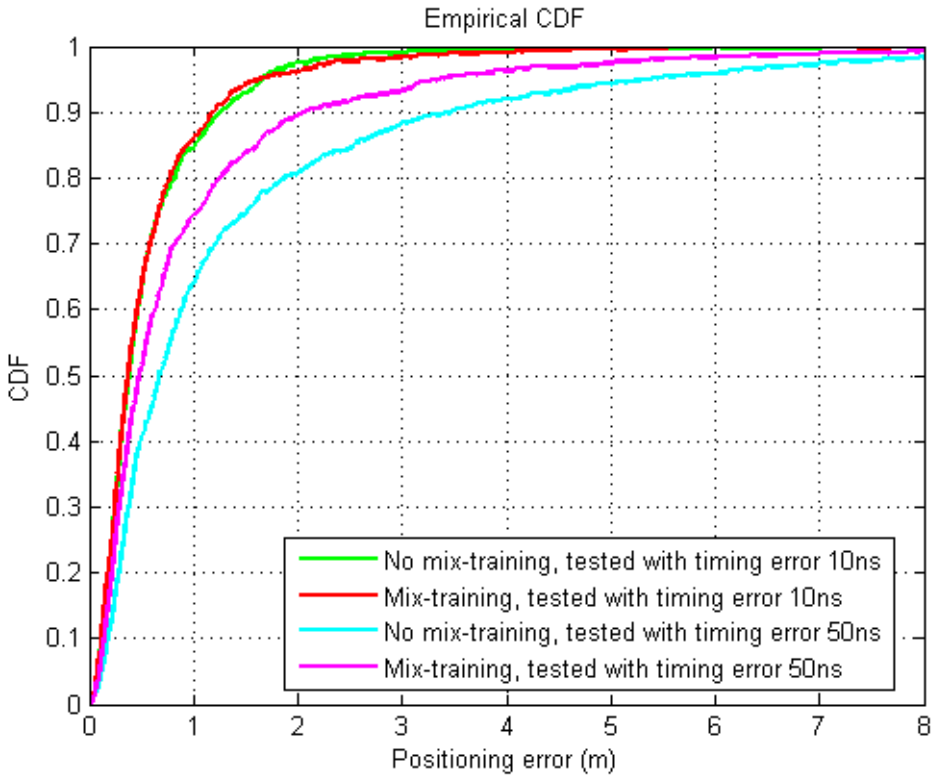


Figure 37 Evaluation of the impact of UE timing error on positioning accuracy

Observation 22: The positioning accuracy of AI/ML based positioning degrades with the increase of network synchronization error for direct AI/ML positioning.

Observation 23: The positioning accuracy of AI/ML model is significantly improved from 3.45m@90% to 2.04m@90% by mix-training with samples of UE timing error for direct AI/ML positioning.

Proposal 10: Further study the impact and potential solution of UE timing error for direct AI/ML positioning.

4.4. The impact of implementation imperfections for AI/ML assisted positioning

In section 4.3, we have evaluated the impact of implementation imperfections for direct AI/ML positioning, including CIR estimation error, synchronization error, labeling error and UE timing error. The simulation results show that these imperfect factors can cause considerable degradation in positioning performance of AI/ML model. Fortunately, the impact of these imperfect factors can be significantly alleviated and even eliminated by data augmentation, such as mix training. For AI/ML assisted positioning, it is expected that these imperfect factors can

also have an adverse impact on the positioning performance. In order to assess the unknown risks from implementations, we evaluate the impact of these imperfect factors on positioning performance for AI/ML assisted positioning, and propose a potential solution to mitigate these impacts as much as possible.

4.4.1. CIR estimation error

Section 4.3.1 confirms that CIR estimation error can severely impair the positioning performance for direct AI/ML positioning. Here, we further evaluate its impact on positioning performance for AI/ML assisted positioning, and the specific simulation assumptions are the same as those for direct AI/ML positioning.

As shown in Table 29, it is indicated that CIR estimation error can dramatically destroy the positioning performance of AI/ML assisted positioning. While compared to direct AI/ML positioning, AI/ML assisted positioning is more robust to small CIR estimation error, such as the case with 1 interfering TRP. Therefore, it is necessary to study the solutions to mitigate the impact of CIR estimation error on positioning performance for AI/ML assisted positioning.

Table 29 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Without interference	0 interfering TRP (Without interference)	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0		1 interfering TRP	25k	1k	4.26M*18	8.50M*18	3.43
CIR	TOA	0		4 interfering TRPs	25k	1k	4.26M*18	8.50M*18	11.35
CIR	TOA	0		8 interfering TRPs	25k	1k	4.26M*18	8.50M*18	16.01

Table 30 CDF of estimation accuracy of intermediate feature TOA (meter), UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Without interference	0 interfering TRP (Without interference)	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0		1 interfering TRP	25k	1k	4.26M*18	8.50M*18	2.76
CIR	TOA	0		4 interfering TRPs	25k	1k	4.26M*18	8.50M*18	9.32
CIR	TOA	0		8 interfering TRPs	25k	1k	4.26M*18	8.50M*18	11.37

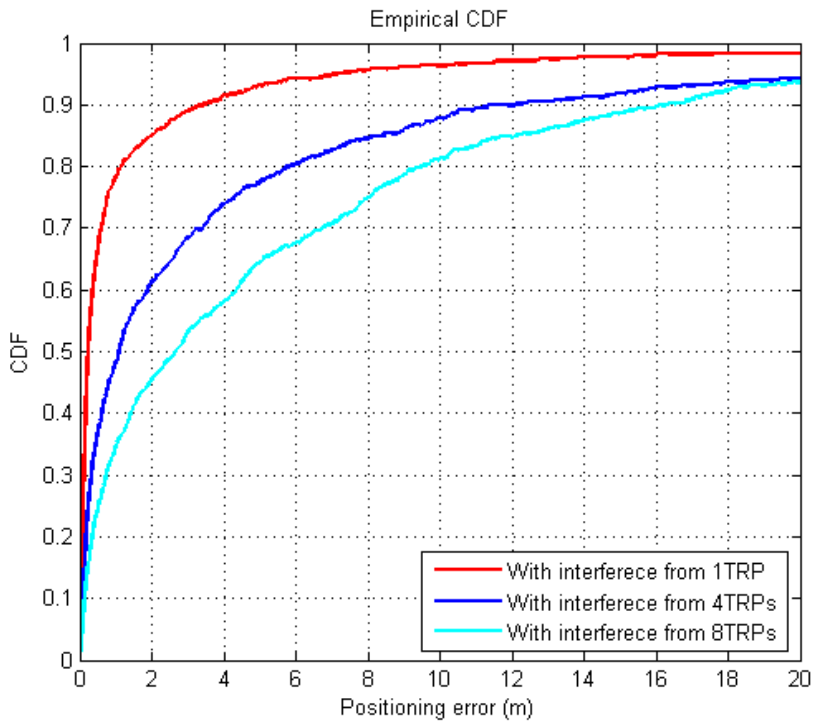


Figure 38 CDF of positioning accuracy

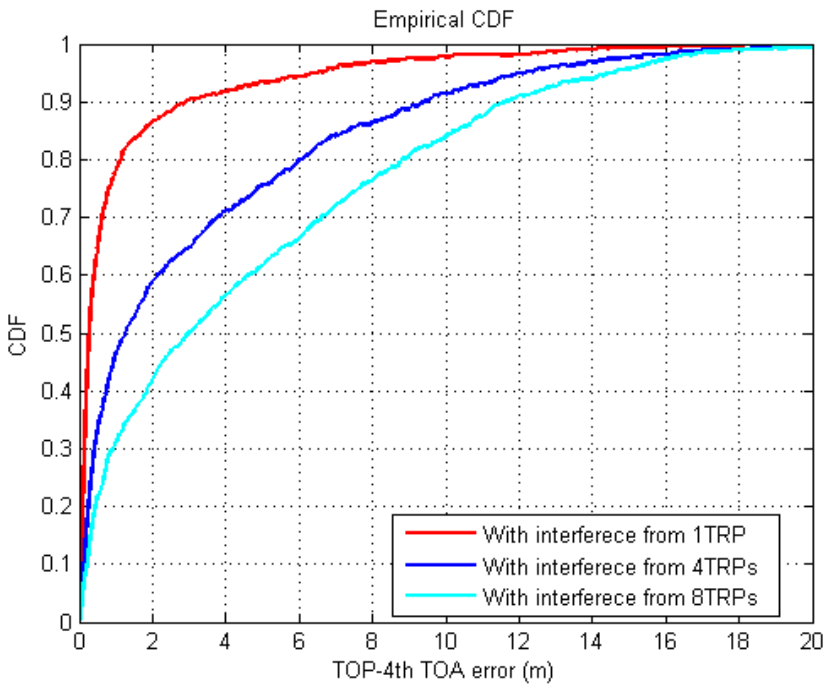


Figure 39 CDF of TOA of positioning accuracy

Observation 24: CIR estimation error can dramatically degrade the positioning performance of AI/ML assisted positioning, while is more robust to small CIR estimation error compared to direct AI/ML positioning.

Proposal 11: Study the solutions to mitigate the impact of CIR estimation error on positioning performance for AI/ML assisted positioning.

4.4.2. Synchronization error

Section 4.3.2 confirms that synchronization error can severely impair the positioning performance for direct AI/ML positioning, which can be greatly alleviated by mix-training or data augmentation. Here, we further

evaluate its impact on positioning performance for AI/ML assisted positioning, and the specific simulation assumptions are the same as those for direct AI/ML positioning.

As shown in Table 31, it is noticeable that synchronization error can dramatically deteriorate the positioning performance of AI/ML model for AI/ML assisted positioning, but it is still better than direct AI/ML positioning as compared to Table 23. Meanwhile, the positioning accuracy significantly degrades with the increase of synchronization error. Therefore, the impact of synchronization error on positioning performance can not be ignored.

Table 31 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	0ns	0ns	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	0ns	2ns	25k	1k	4.26M*18	8.50M*18	1.63
CIR	TOA	0	0ns	10ns	25k	1k	4.26M*18	8.50M*18	2.05
CIR	TOA	0	0ns	50ns	25k	1k	4.26M*18	8.50M*18	8.45

Table 32 CDF of estimation accuracy of intermediate feature TOA (meter), UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	0ns	0ns	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	0ns	2ns	25k	1k	4.26M*18	8.50M*18	1.46
CIR	TOA	0	0ns	10ns	25k	1k	4.26M*18	8.50M*18	1.70
CIR	TOA	0	0ns	50ns	25k	1k	4.26M*18	8.50M*18	6.45

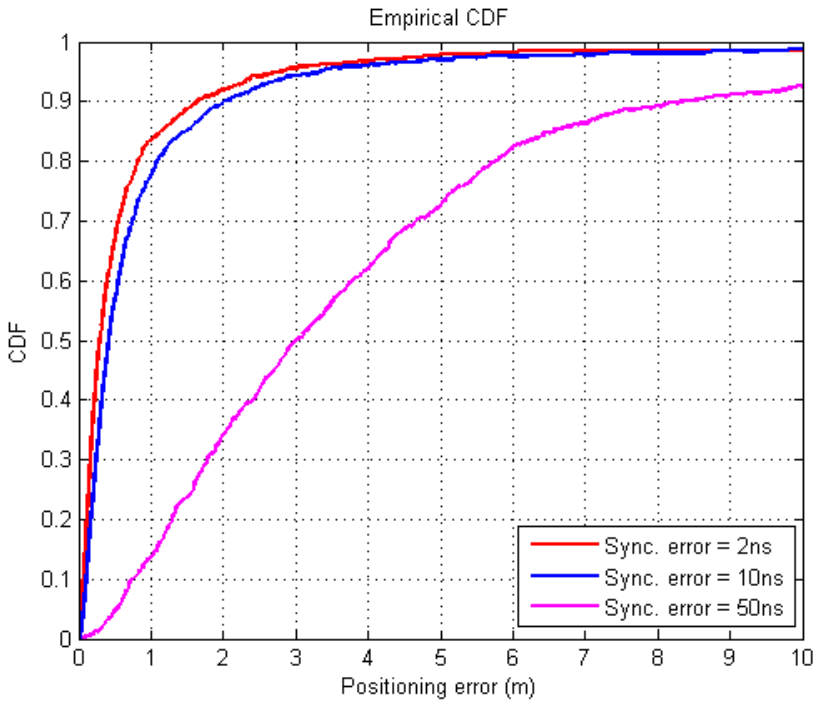


Figure 40 CDF of positioning accuracy

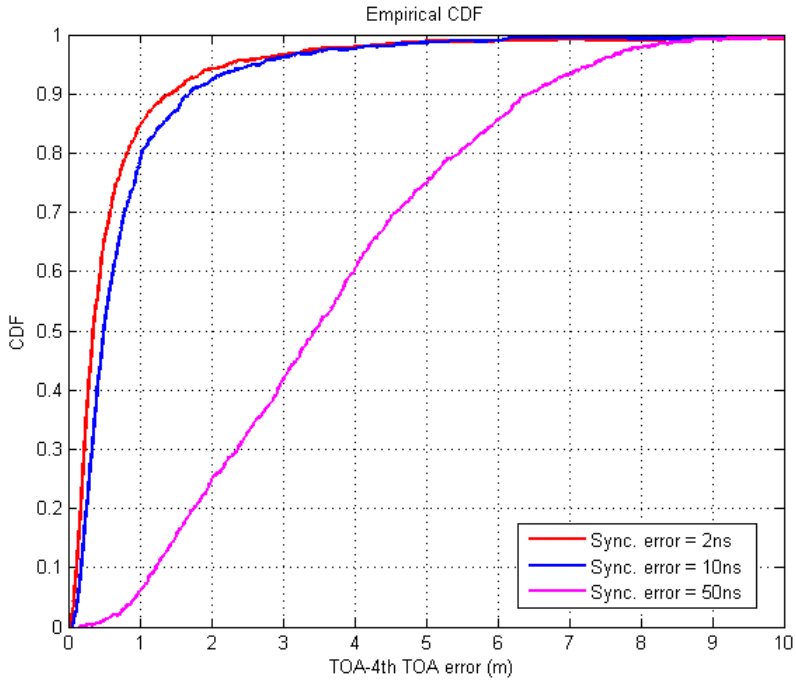


Figure 41 CDF of TOA of positioning accuracy

Observation 25: The positioning accuracy of AI/ML assisted positioning significantly degrades with the increase of network synchronization error, but it is still better than direct AI/ML positioning.

Proposal 12: Study the solutions to mitigate the impact of network synchronization error on positioning performance for AI/ML assisted positioning.

4.4.3. Labeling error

Section 4.3.3 confirms that labeling error of data collection can severely impair the positioning performance for direct AI/ML positioning. Here, we further evaluate its impact on positioning performance for AI/ML assisted positioning, and the specific simulation assumptions are same as those for direct AI/ML positioning.

As shown in Table 33, it is observed that AI/ML assisted positioning is less sensitive to labeling error compared with direct AI/ML positioning. For example, for the case STD = 2m, the positioning accuracy of AI/ML assisted positioning is 2.24m@90%, greatly higher than that of direct AI/ML positioning 3.55m@90%. This robustness is partly illustrated as Figure 42. For a ground truth label with labeling error (e_x, e_y), the corresponding TOA labeling error is given by

$$\tilde{e} = \left|\tilde{d} - d\right| = \left|\sqrt{d^2 + e^2 - 2de\cos(\theta)} - d\right| \le e$$

Where $e = \sqrt{e_x^2 + e_y^2}$. In general, the labeling error of TOA is smaller than that of location, i.e., $\tilde{e} \le e$.

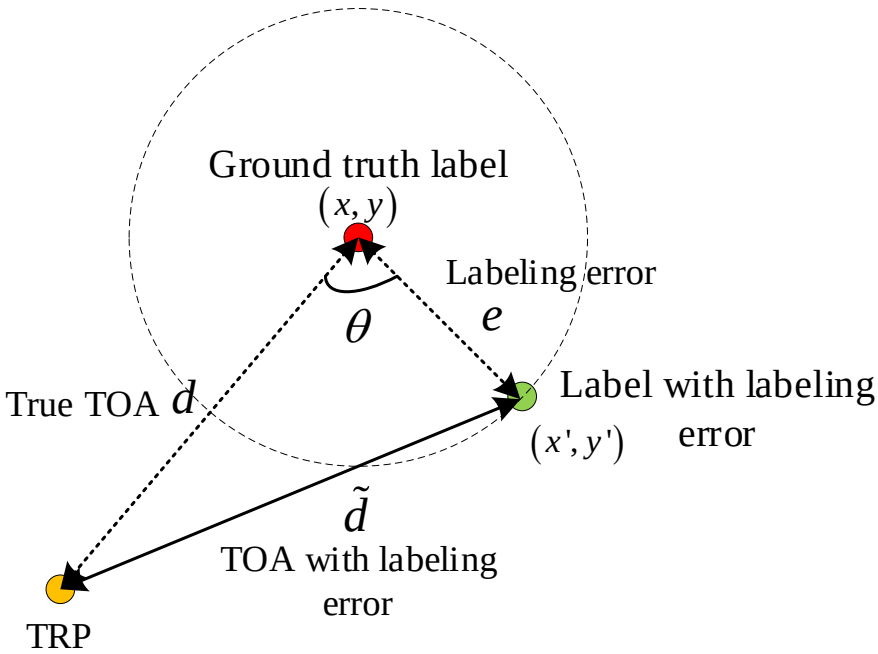


Figure 42 Illustration of TOA labeling error

Table 33 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Std = 0	0	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	Std = 0.5	0	25k	1k	4.26M*18	8.50M*18	0.90
CIR	TOA	0	Std = 1	0	25k	1k	4.26M*18	8.50M*18	1.73
CIR	TOA	0	Std = 2	0	25k	1k	4.26M*18	8.50M*18	2.24

Table 34 CDF of estimation accuracy of intermediate feature TOA (meter), UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	Std = 0	0	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	Std = 0.5	0	25k	1k	4.26M*18	8.50M*18	1.02

CIR	TOA	0	Std = 1	0	25k	1k	4.26M*18	8.50M*18	1.61
CIR	TOA	0	Std = 2	0	25k	1k	4.26M*18	8.50M*18	2.20

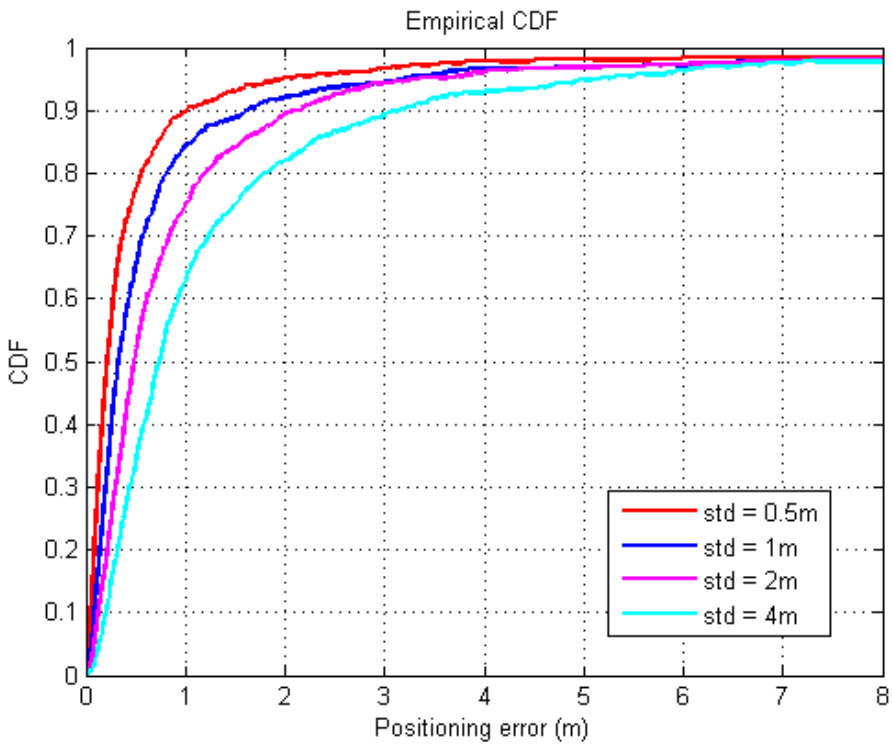


Figure 43 CDF of positioning accuracy

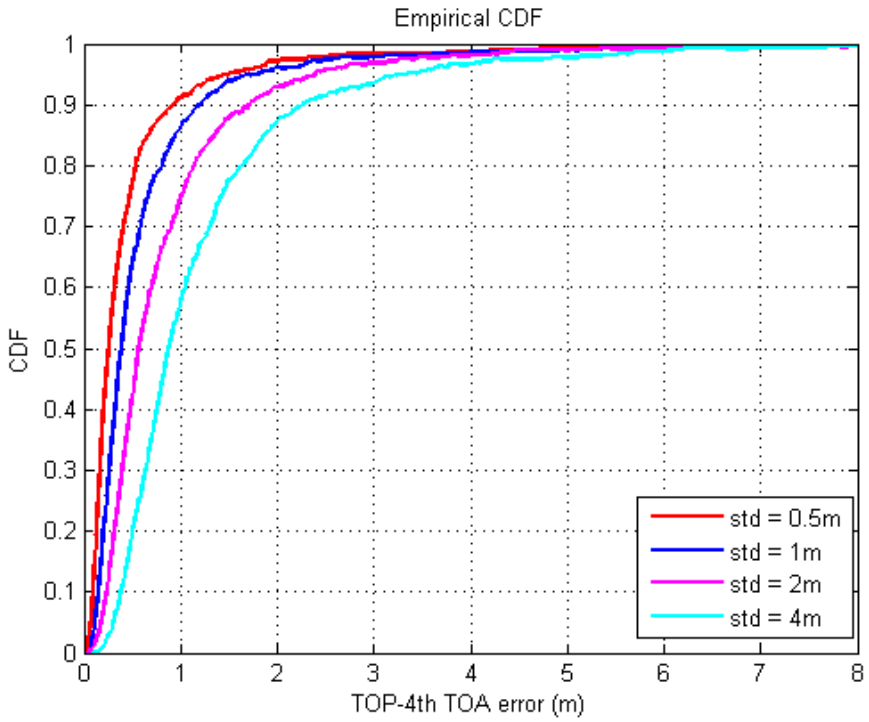


Figure 44 CDF of TOA of positioning accuracy

Observation 26: Labeling error can slightly impair the positioning performance for AI/ML assisted positioning, and AI/ML assisted positioning is more robust to labeling error compared with direct AI/ML positioning.

Proposal 13: Study the solutions to mitigate the impact of labeling error on positioning performance for AI/ML assisted positioning.

4.4.4. UE timing error

Section 4.3.4 confirms that UE timing error can severely impair the positioning performance for direct AI/ML positioning, which can be greatly alleviated by mix-training or data augmentation. Here, we further evaluate its impact on positioning performance for AI/ML assisted positioning, and the specific simulation assumptions are the same as those for direct AI/ML positioning.

As shown in Table 35, it is noticeable that large UE timing error can dramatically deteriorate the positioning performance of AI/ML model for AI/ML assisted positioning. For example, the positioning accuracy degrades from 0.73m@90% to 10.18m@90% when UE timing error is 50ns. However, UE timing error can slightly impair the positioning performance when the timing error is less than 10ns. Meanwhile, the positioning accuracy gradually degrades with the increase of UE timing error. Therefore, the impact of UE timing error can not be ignored, and it is necessary to study the solutions to mitigate the impact of UE timing error on positioning performance.

Table 35 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	0ns	0ns	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	0	0ns	2ns	25k	1k	4.26M*18	8.50M*18	0.75
CIR	TOA	0	0ns	10ns	25k	1k	4.26M*18	8.50M*18	1.51
CIR	TOA	0	0ns	50ns	25k	1k	4.26M*18	8.50M*18	10.18

Table 36 CDF of estimation accuracy of intermediate feature TOA (meter), UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0	0ns	0ns	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	0	0ns	2ns	25k	1k	4.26M*18	8.50M*18	0.64
CIR	TOA	0	0ns	10ns	25k	1k	4.26M*18	8.50M*18	1.39
CIR	TOA	0	0ns	50ns	25k	1k	4.26M*18	8.50M*18	9.00

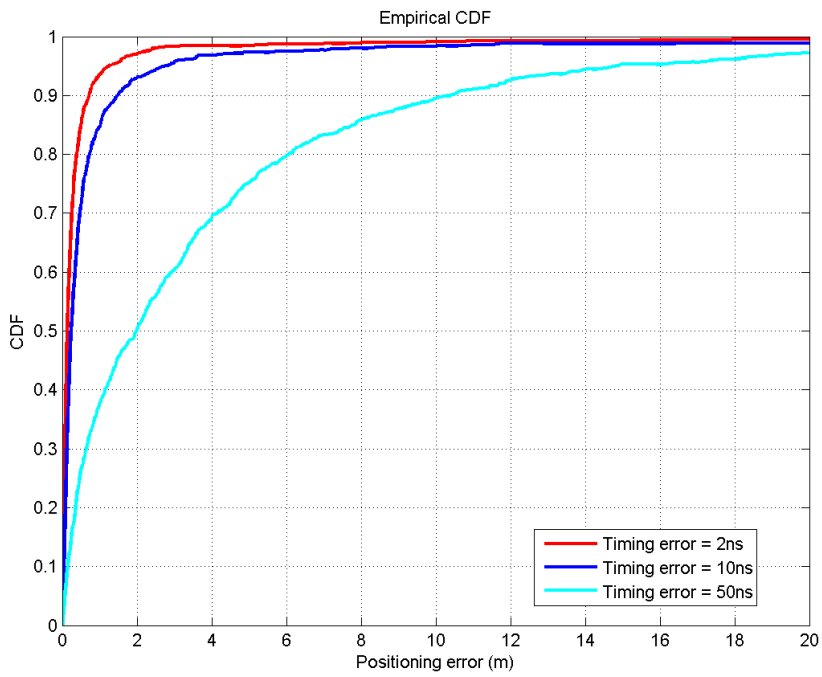


Figure 45 CDF of positioning accuracy

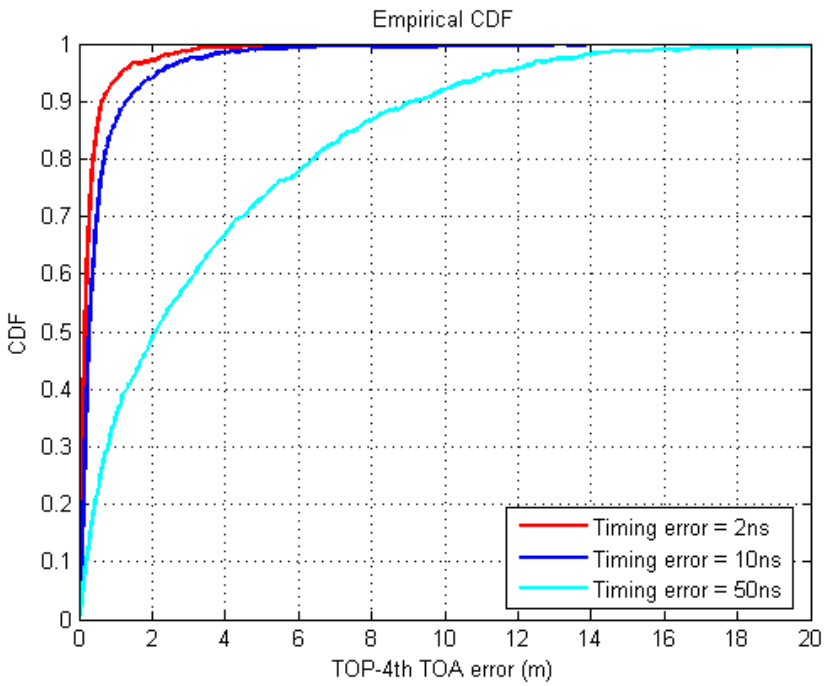


Figure 46 CDF of TOA of positioning accuracy

Observation 27: Large UE timing error can dramatically deteriorate the positioning performance of AI/ML model for AI/ML assisted positioning, such as 50ns.

Proposal 14: Study the solutions to mitigate the impact of UE timing error on positioning performance for AI/ML assisted positioning.

5. Model fine-tuning for generalization enhancement

At the RAN1#110 meeting, it was agreed that:

Agreement

For AI/ML-based positioning, for evaluation of the potential performance benefits of model finetuning, report at least the following:

- training dataset setting (e.g., training dataset size necessary for performing model finetuning)
- horizontal positioning accuracy (in meters) before and after model finetuning.

At the RAN1#111 meeting, it was agreed that:

Agreement

For both direct and AI/ML assisted positioning methods, investigate at least the impact of the amount of fine-tuning data on the positioning accuracy of the fine-tuned model.

- The fine-tuning data is the training dataset from the target deployment scenario.

Following above agreements, we further evaluate the impact of the amount of fine-tuning data on the positioning accuracy of the fine-tuned model for direct AI/ML positioning and AI/ML assisted positioning.

When AI model trained offline is transferred to a new scenario, performance degradation may be inevitable due to the mismatch between training data and field data as shown in the above sections. In general, there are two solutions to deal with this generalization problem:

- The first is to ensure training data and field data are sampled from the same scenario. In this way, the network entity or UE needs to collect large amounts of data for model training and validation, and considerable computational and time resource are also required to train these scenario-specific models from scratch.
- The second is fine-tuning. Specifically, AI model is pre-trained by training data which may be from simulation data, field data collected by other drops, or both. When the pre-trained model is transferred to a real environment, a retraining process, named fine-tuning, should be triggered to fine-tune the pre-trained model with field data collected from the real environment. In this way, a scenario-specific model can be obtained with a small amount of field data and computation & time resource consumption.

The first solution is obvious and intuitive, which can achieve good performance at the cost of heavy data collection and model retraining. In this section, we mainly focus on the second solution utilizing fine-tuning to enhance the model generalization performance. In general, fine-tuning procedure consists of two steps. The first step is to pre-train a model based on some offline-collected data. The second step is to fine-tune the pre-trained model based on the collected field data.

Before applying fine-tuning, at least the following issues should be identified and resolved firstly, including：

- What scenarios or tasks are model fine-tuning applied to? Both original domain and target domain should be identified.
- How many field samples are required to conduct fine-tuning? Some guidelines on sample size should be considered.

To answer above questions, we perform the following simulation evaluation and analysis, and some interesting and meaningful observations are also presented.

5.1. Model fine-tuning for direct AI/ML positioning

From the observations in Section 4, we have concluded that direct AI/ML positioning suffers from poor generalization performance for different drops, clutter parameters, scenarios and synchronization errors. Here, we will evaluate whether model fine-tuning can improve the generalization performance for direct AI/ML positioning.

5.1.1. Model fine-tuning across clutter parameters

When the offline-trained AI/ML model is deployed in a scenario with a different clutter parameter, positioning performance degradation is unavoidable. Fortunately, fine-tuning can be a useful technique to mitigate the impact of these environmental changes. As shown in Table 37, we can observe that fine-tuning the model with a small amount of field data can significantly improve the positioning accuracy in the new scenario with clutter parameter {0.4, 2, 2}. Moreover, the positioning accuracy continues to improve as the increased size of the field data used for model fine-tuning.

Table 37 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)	Dataset size	AI/ML complexity	Horizontal pos. accuracy

											at CDF=90 % (m)
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	{0.6, 6, 2}	/	{0.4, 2, 2}	25k	0	1k	1.65M	22.30M	8.67
CIR	Pos.	0%	{0.4, 2, 2}	/	{0.6, 6, 2}	25k	0	1k	1.65M	22.30M	4.77
CIR	Pos.	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	0.5k	1k	1.65M	22.30M	5.22
CIR	Pos.	0%	{0.4, 2, 2}	{0.6, 6, 2}	{0.6, 6, 2}	25k	0.5k	1k	1.65M	22.30M	3.89
CIR	Pos.	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	1k	1.65M	22.30M	4.40
CIR	Pos.	0%	{0.4, 2, 2}	{0.6, 6, 2}	{0.6, 6, 2}	25k	1k	1k	1.65M	22.30M	3.23
CIR	Pos.	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	2k	1k	1.65M	22.30M	3.50
CIR	Pos.	0%	{0.4, 2, 2}	{0.6, 6, 2}	{0.6, 6, 2}	25k	2k	1k	1.65M	22.30M	2.56
CIR	Pos.	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	3k	1k	1.65M	22.30M	3.16
CIR	Pos.	0%	{0.4, 2, 2}	{0.6, 6, 2}	{0.6, 6, 2}	25k	3k	1k	1.65M	22.30M	2.40

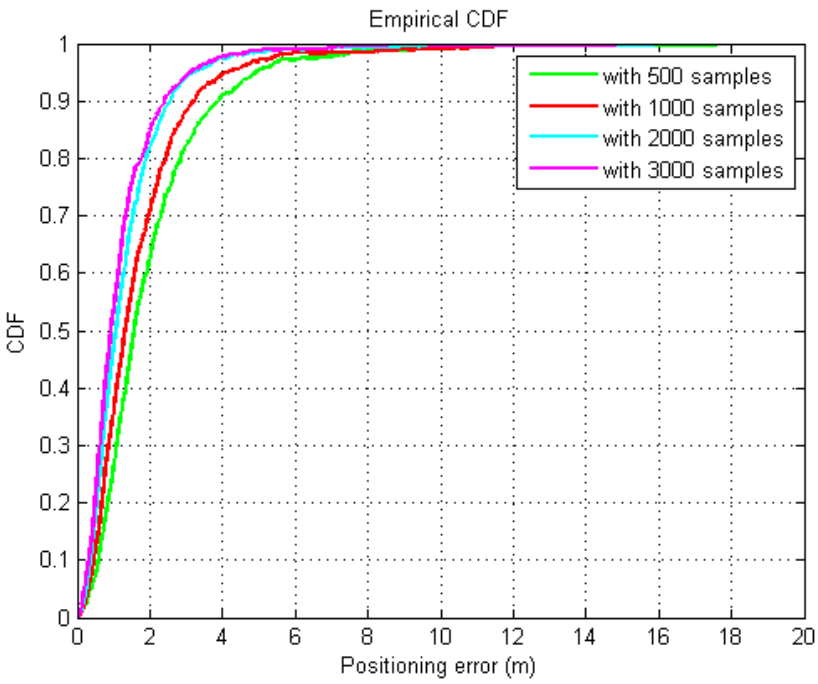


Figure 47 Evaluation of model fine-tuning for different clutter parameters (train with {0.4, 2, 2}, fine-tuning and testing with {0.6, 6, 2})

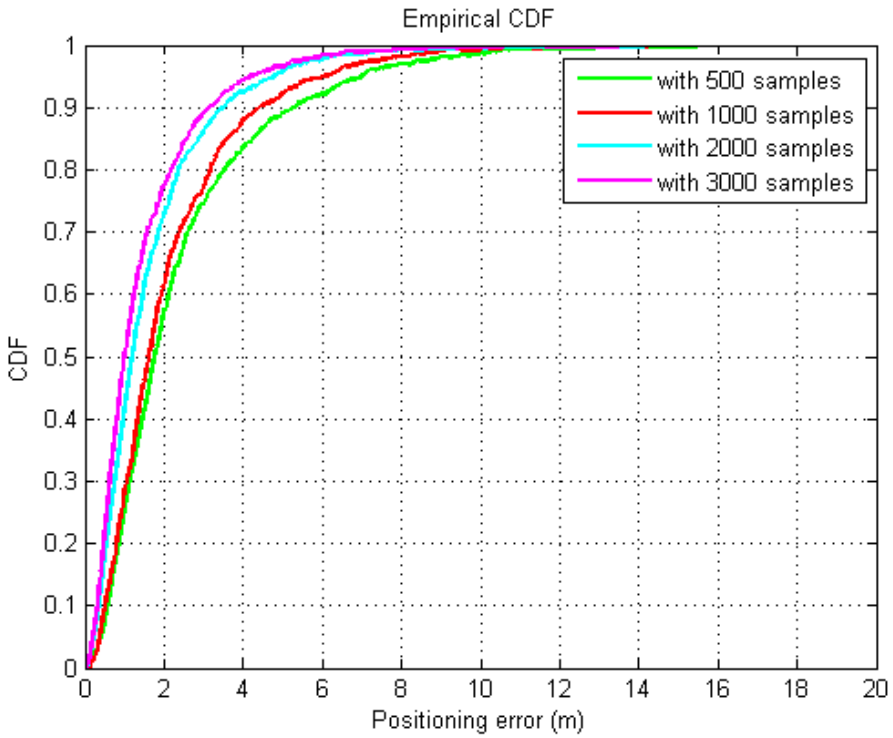


Figure 48 Evaluation of model fine-tuning for different clutter parameters (train with {0.6, 6, 2}, fine-tuning and testing with {0.4, 2, 2}), UE distribution area = [120x60 m]

Observation 28: Fine-tuning the model with small amounts of samples from an unseen clutter parameter configuration can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such clutter parameter for direct AI/ML positioning.

5.1.2. Model fine-tuning across drops

When the AI/ML model offline trained with Drop1 is deployed in Drop2, obvious positioning performance degradation has been observed. As shown in Table 38, we can observe that fine-tuning the model with a small amount of the field data can significantly improve the positioning accuracy in the new drop. Moreover, the positioning accuracy continues to improve as the increased size of the field data used for model fine-tuning.

Table 38 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90 % (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	Drop1	/	Drop 2	25k	0	1k	1.65M	22.30M	6.00
CIR	Pos.	0%	Drop1	Drop 2	Drop 2	25k	0.5k	1k	1.65M	22.30M	4.69
CIR	Pos.	0%	Drop1	Drop 2	Drop 2	25k	1k	1k	1.65M	22.30M	3.97
CIR	Pos.	0%	Drop1	Drop 2	Drop 2	25k	2k	1k	1.65M	22.30M	3.37
CIR	Pos.	0%	Drop1	Drop 2	Drop 2	25k	3k	1k	1.65M	22.30M	2.90

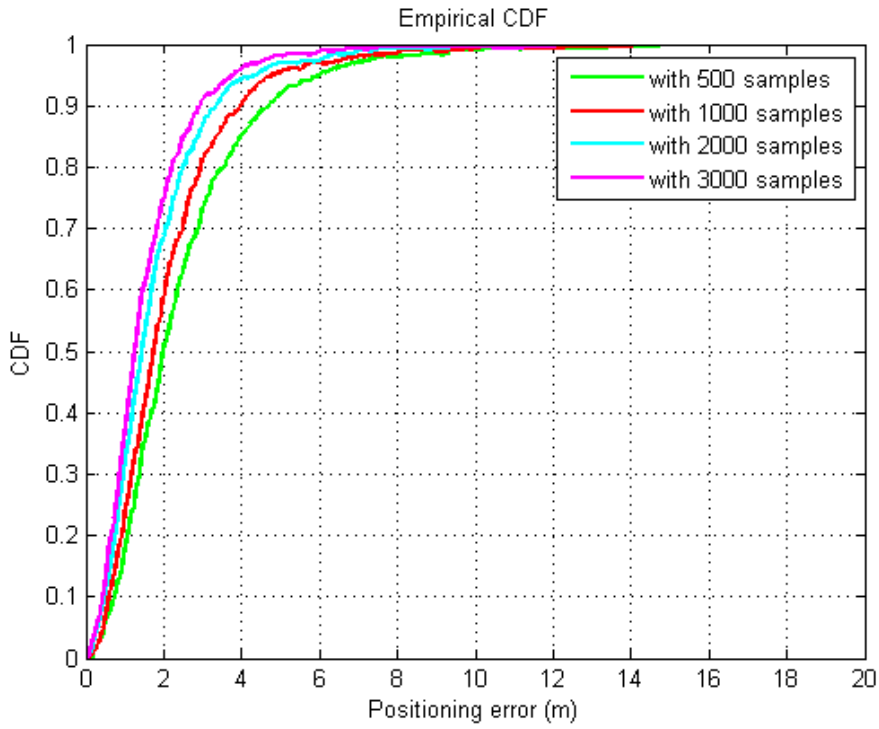


Figure 49 Evaluation of model fine-tuning for different drops

Observation 29: Fine-tuning the model with small amounts of samples from an unseen drop can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to such new drop for direct AI/ML positioning.

5.1.3. Model fine-tuning across scenarios

When the offline-trained AI/ML model is deployed in a different scenario, positioning performance degradation is inevitable. As shown in Table 39, fine-tuning the model with a small amount of the field data can significantly improve the positioning accuracy in the new scenario. Moreover, the positioning accuracy continues to improve as the increase of the field data used for model fine-tuning.

Table 39 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90 % (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	DH	/	HH	25k	0	1k	1.65M	22.30M	>10
CIR	Pos.	0%	DH	HH	HH	25k	0.5k	1k	1.65M	22.30M	10.50
CIR	Pos.	0%	DH	HH	HH	25k	1k	1k	1.65M	22.30M	8.78
CIR	Pos.	0%	DH	HH	HH	25k	2k	1k	1.65M	22.30M	5.84
CIR	Pos.	0%	DH	HH	HH	25k	3k	1k	1.65M	22.30M	4.66

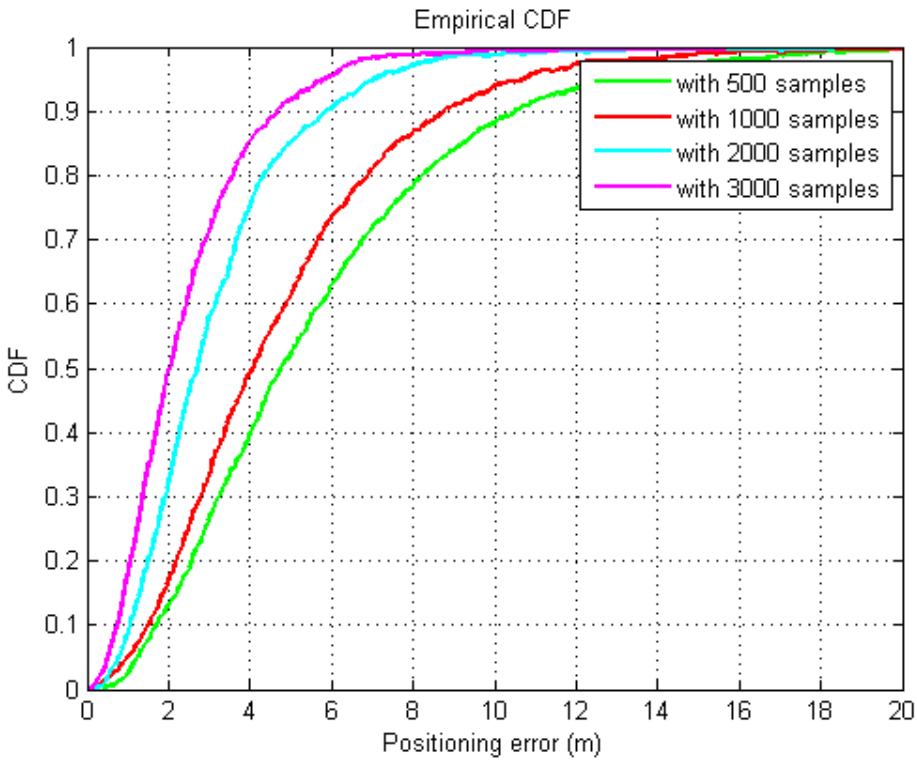


Figure 50 Evaluation of model fine-tuning for different scenarios

Observation 30: Fine-tuning the model with small amounts of samples from an unseen scenario can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario for direct AI/ML positioning.

5.1.4. Model fine-tuning across synchronization errors

Network synchronization error is inevitable, and can severely deteriorate the positioning accuracy of AI/ML model. In this regard, we expect to mitigate the negative impact of network synchronization error by model fine-tuning. Assuming that the initial AI/ML model is pretrained with offline-collected data without synchronization error, model fine-tuning is performed with collected field data with actual synchronization error when the pretrained AI/ML model is transferred or deployed in a practical scenario. As shown in Table 40, Table 41 and Table 42, we evaluate the gain of model fine-tuning for scenarios with 50ns, 10ns and 2ns synchronization errors, respectively. Simulation results indicate that model fine-tuning can significantly improve the positioning accuracy of AI/ML model with a small amount of field data. In particular, fine-tuning the AI/ML model with only 3000 samples can achieve comparable positioning accuracy as compared with large-scale model training with 25k synchronization error-free data (0.99m@90%). Moreover, the positioning accuracy continues to improve with the increase of field data. Therefore, it is concluded that model fine-tuning can obviously mitigate the impact of synchronization errors for direct AI/ML positioning.

Table 40 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90 % (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	Sync. 0ns	/	50ns	25k	0	1k	1.65M	22.30M	10.18
CIR	Pos.	0%	0ns	50ns	50ns	25k	0.5k	1k	1.65M	22.30M	3.22

CIR	Pos.	0%	0ns	50ns	50ns	25k	1k	1k	1.65M	22.30M	2.39
CIR	Pos.	0%	0ns	50ns	50ns	25k	2k	1k	1.65M	22.30M	1.73
CIR	Pos.	0%	0ns	50ns	50ns	25k	3k	1k	1.65M	22.30M	1.47

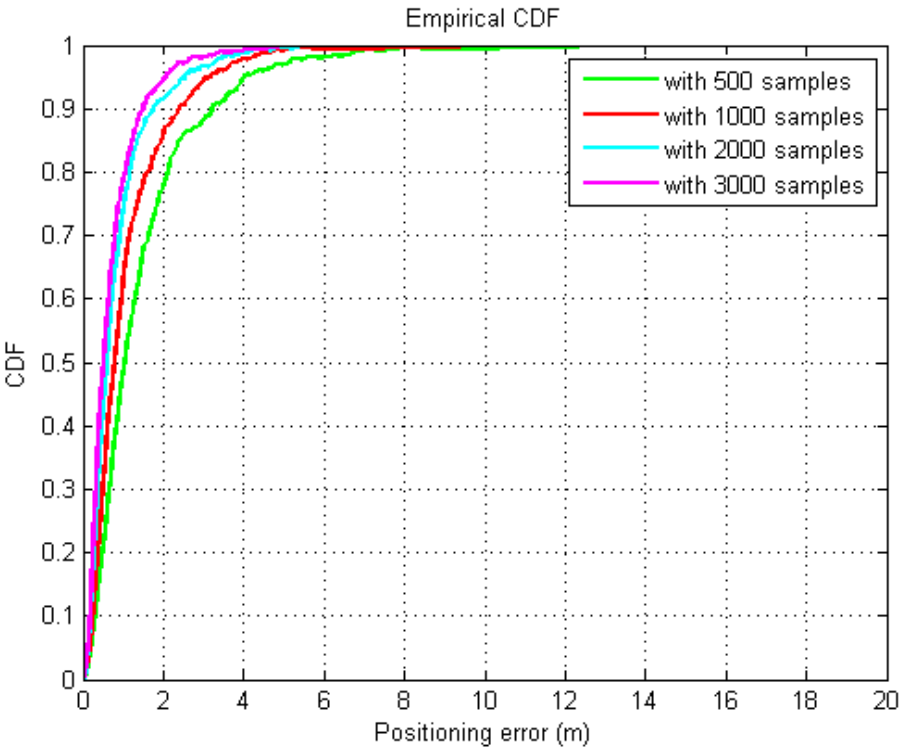


Figure 51 Evaluation of model fine-tuning for different synchronization errors (50ns)

Table 41 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90 % (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	Sync. 0ns	/	10ns	25k	0	1k	1.65M	22.30M	4.56
CIR	Pos.	0%	0ns	10ns	10ns	25k	0.5k	1k	1.65M	22.30M	1.44
CIR	Pos.	0%	0ns	10ns	10ns	25k	1k	1k	1.65M	22.30M	1.28
CIR	Pos.	0%	0ns	10ns	10ns	25k	2k	1k	1.65M	22.30M	1.06
CIR	Pos.	0%	0ns	10ns	10ns	25k	3k	1k	1.65M	22.30M	0.95

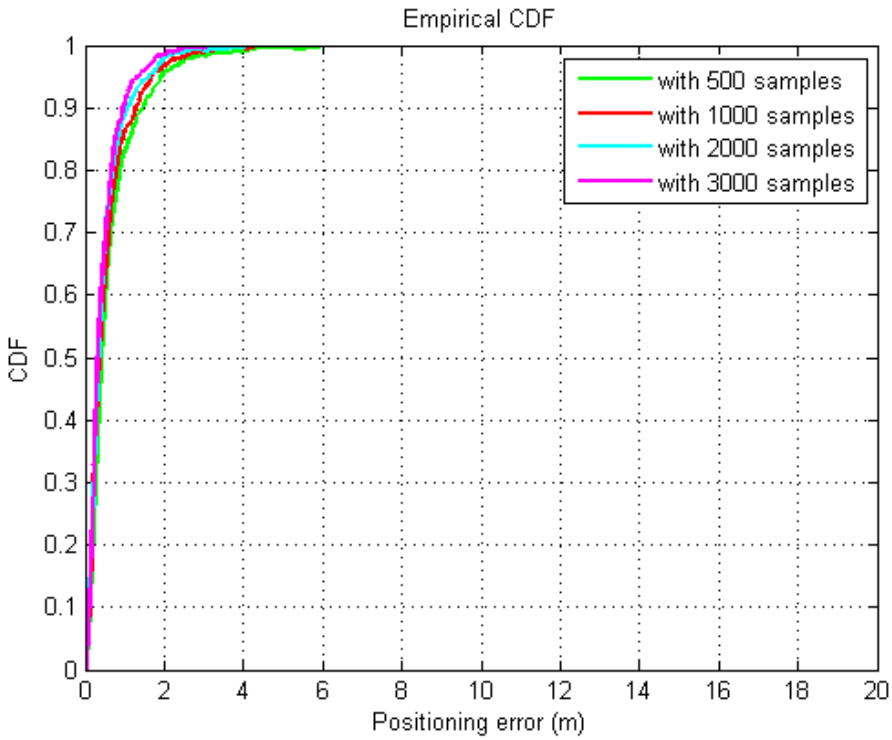


Figure 52 Evaluation of model fine-tuning for different synchronization errors (10ns)

Table 42 Evaluation results for AI/ML model deployed on UE or Network side, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90 % (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	0%	Sync. 0ns	/	2ns	25k	0	1k	1.65M	22.30M	1.64
CIR	Pos.	0%	0ns	2ns	2ns	25k	0.5k	1k	1.65M	22.30M	1.11
CIR	Pos.	0%	0ns	2ns	2ns	25k	1k	1k	1.65M	22.30M	1.11
CIR	Pos.	0%	0ns	2ns	2ns	25k	2k	1k	1.65M	22.30M	0.95
CIR	Pos.	0%	0ns	2ns	2ns	25k	3k	1k	1.65M	22.30M	0.90

Observation 31: Fine-tuning the model with small amounts of samples with an unseen synchronization error can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such synchronization error for direct AI/ML positioning.

Proposal 15: Further study and confirm the benefits of fine-tuning in terms of model generalization enhancement for direct AI/ML positioning.

5.2. Model fine-tuning for AI/ML assisted positioning

From the observations in Section 4, we have concluded that AI/ML assisted positioning suffers from poor generalization performance in some cases. Here, we will evaluate whether model fine-tuning can improve the generalization performance for AI/ML assisted positioning. For comparison, AI/ML based TOA estimation method without model fine-tuning is adopted as the baseline.

5.2.1. Model fine-tuning across clutter parameters

We firstly evaluate the benefits of model fine-tuning in terms of improving model generalization capability across clutter parameters. As shown in Table 43, fine-tuning the model with only 1k samples can achieve huge positioning accuracy enhancement from 3.7m@90% to 0.63m@90% when the AI/ML model pretrained with DH{0.6, 6, 2} is transferred to DH{0.4, 2, 2} as compared with no model fine-tuning. Furthermore, with the increase of samples used for model fine-tuning, the positioning accuracy of AI/ML model gradually improves from 0.85m(1k)@90% to 0.48m(3k)@90% under the unseen clutter parameter configuration.

Table 43 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal Pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	{0.6, 6, 2}	/	{0.4, 2, 2}	25k	0	1k	4.26M*18	8.50M*18	3.70
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	0.5k	1k	4.26M*18	8.50M*18	0.85
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	1k	4.26M*18	8.50M*18	0.63
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	2k	1k	4.26M*18	8.50M*18	0.48
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	3k	1k	4.26M*18	8.50M*18	0.48

Table 44 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	{0.6, 6, 2}	/	{0.4, 2, 2}	25k	0	1k	4.26M*18	8.50M*18	1.49
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	0.5k	1k	4.26M*18	8.50M*18	0.36
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	1k	1k	4.26M*18	8.50M*18	0.33
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	2k	1k	4.26M*18	8.50M*18	0.27
CIR	TOA	0%	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	25k	3k	1k	4.26M*18	8.50M*18	0.27

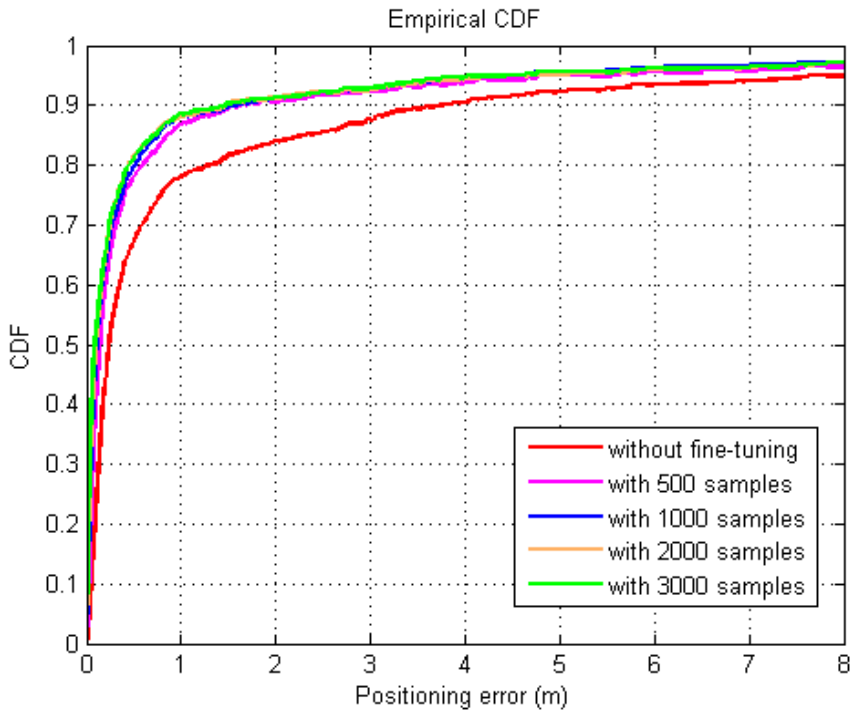


Figure 53 Positioning accuracy of model fine-tuning for different clutter parameters (train with {0.4, 2, 2}, fine-tuning and testing with {0.6, 6, 2})

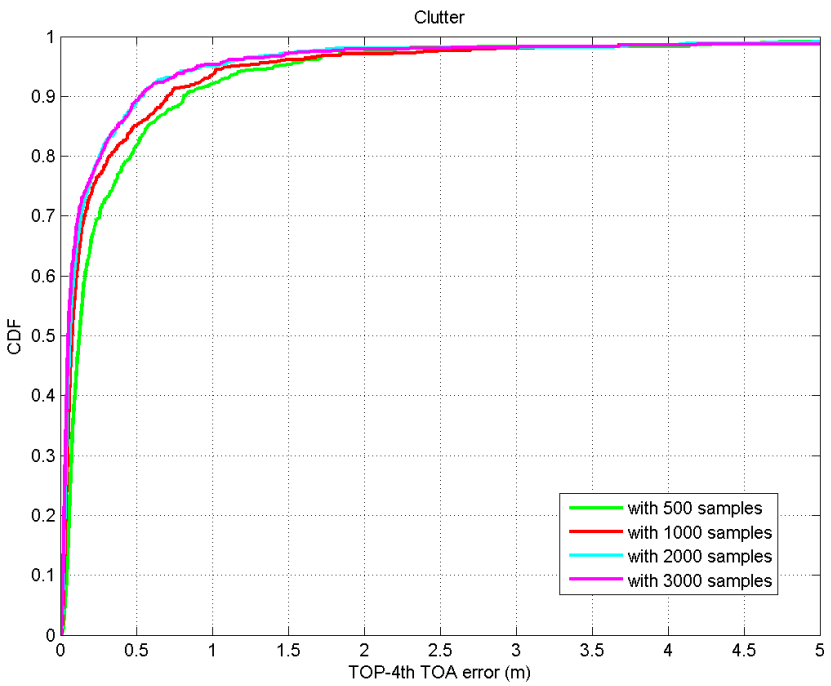


Figure 54 TOA accuracy of model fine-tuning for different clutter parameters (train with {0.4, 2, 2}, fine-tuning and testing with {0.6, 6, 2})

Observation 32: Fine-tuning the model with small amounts of samples from an unseen clutter parameter can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such clutter parameter for AI/ML assisted positioning.

5.2.2. Model fine-tuning across drops

We evaluate the benefits of model fine-tuning in terms of improving model generalization capability across drops. As shown in Table 45, fine-tuning the model with limited samples can achieve obvious positioning accuracy enhancement when the AI/ML model pretrained with the data of drop 1 is transferred to another drop compared

to no model fine-tuning. Moreover, with the increase of samples used for model fine-tuning, the positioning accuracy of AI/ML model gradually improves but is still worse than that of fine-tuning the model of DH{0.6, 6, 2}with DH{0.4, 2, 2}. The reason behind is that fine-tuning the model with small amounts of samples would not completely capture the fingerprint features of the new drop. Thus, it is suggested that the large-scale dataset is still required to fine-tune the pretrained model to the new environment for fingerprint based positioning, but has the advantages of reduced computational complexity compared with training an AI/ML model from scratch.

Table 45 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal Pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Drop1	/	Drop2	25k	0	1k	4.26M*18	8.50M*18	10.37
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	0.5k	1k	4.26M*18	8.50M*18	5.61
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	1k	1k	4.26M*18	8.50M*18	5.50
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	2k	1k	4.26M*18	8.50M*18	5.03
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	3k	1k	4.26M*18	8.50M*18	4.08

Table 46 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Drop1	/	Drop2	25k	0	1k	4.26M*18	8.50M*18	8.00
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	0.5k	1k	4.26M*18	8.50M*18	5.68
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	1k	1k	4.26M*18	8.50M*18	4.78
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	2k	1k	4.26M*18	8.50M*18	4.42
CIR	TOA	0%	Drop1	Drop2	Drop2	25k	3k	1k	4.26M*18	8.50M*18	3.95

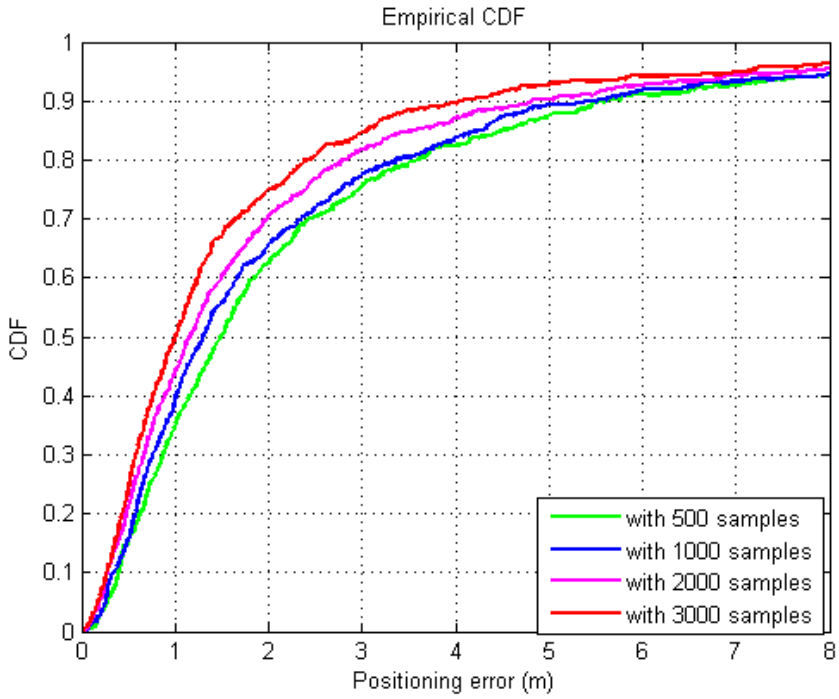


Figure 55 CDF of positioning accuracy of fine-tuning in different drops

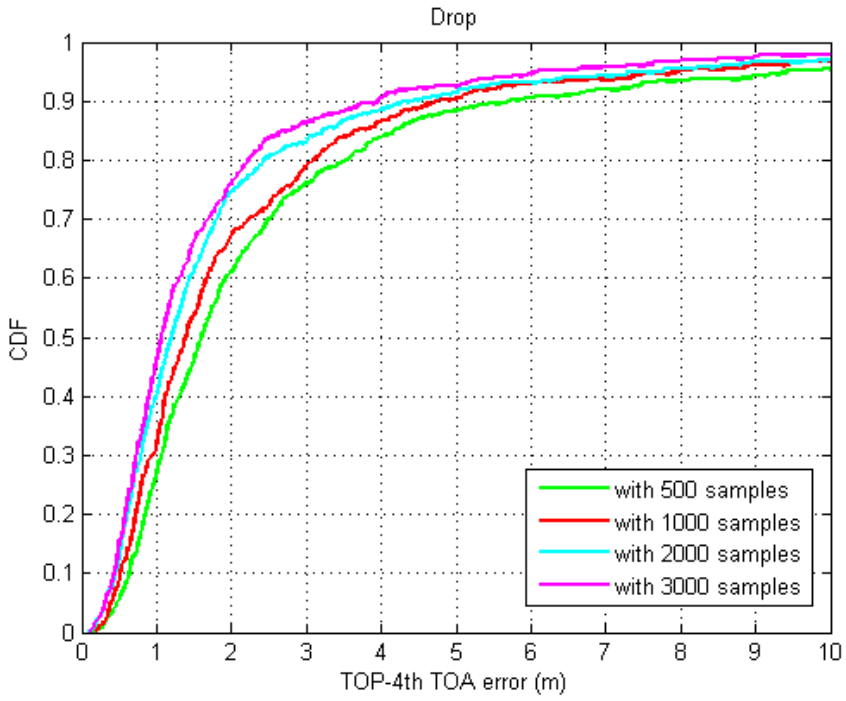


Figure 56 CDF of TOA accuracy of fine-tuning in different drops

Observation 33: Fine-tuning the model with small amounts of samples from an unseen drop can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to such new drop for AI/ML assisted positioning.

Observation 34: The large-scale dataset is still required to fine-tune the pretrained model to the new environment for fingerprint based positioning, but has the advantages of reduced computational complexity compared with training an AI/ML model from scratch.

5.2.3. Model fine-tuning across scenarios

We further evaluate the positioning performance when the pre-trained model trained with InF-DH {0.6, 6, 2} data is transferred to an InF-HH scenario. As listed in Table 47 and Table 49, we can observe that the AI/ML model

trained with InF-DH data would not work in InF-HH and InF-SH scenarios, and the positioning errors are unacceptable (>10m) due to the different distributions of TRPs. However, when the AI/ML model trained with InF-DH data is fine-tuned with only 0.5k samples of InF-HH or SH data, we observe a huge performance improvement from >10m@90% to 0.3m@90%. For AI/ML assisted positioning, the positioning accuracy of fine-tuning the DH model with SH or HH data is greatly better than that of direct AI/ML positioning as presented in Table 39. Note that an AI/ML model trained based on dataset from a scenario may be fine-tuned and then used for a new deployment scenario when there's only limited training data with label (i.e. ground truth UE location) for that new scenario.

Table 47 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal Pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	DH	/	HH	25k	0	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0%	DH	HH	HH	25k	0.5k	1k	4.26M*18	8.50M*18	0.30
CIR	TOA	0%	DH	HH	HH	25k	1k	1k	4.26M*18	8.50M*18	0.17
CIR	TOA	0%	DH	HH	HH	25k	2k	1k	4.26M*18	8.50M*18	0.09
CIR	TOA	0%	DH	HH	HH	25k	3k	1k	4.26M*18	8.50M*18	0.06

Table 48 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	DH	/	HH	25k	0	1k	4.26M*18	8.50M*18	13.56
CIR	TOA	0%	DH	HH	HH	25k	0.5k	1k	4.26M*18	8.50M*18	0.17
CIR	TOA	0%	DH	HH	HH	25k	1k	1k	4.26M*18	8.50M*18	0.09
CIR	TOA	0%	DH	HH	HH	25k	2k	1k	4.26M*18	8.50M*18	0.05
CIR	TOA	0%	DH	HH	HH	25k	3k	1k	4.26M*18	8.50M*18	0.04

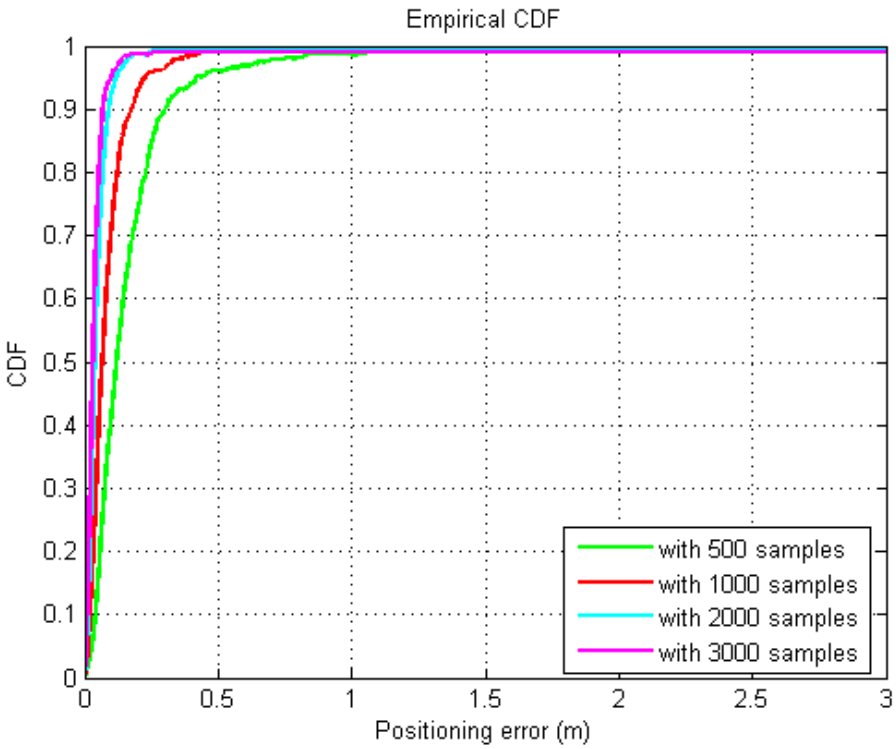


Figure 57 Positioning accuracy of model fine-tuning for different scenarios (train with DH, fine-tuning and testing with HH)

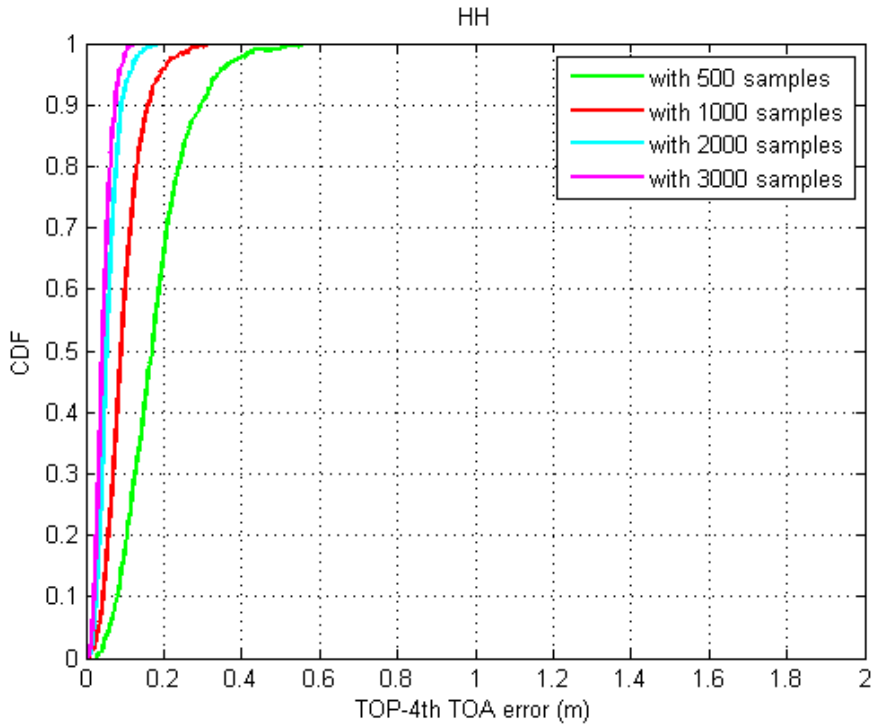


Figure 58 TOA accuracy of model fine-tuning for different scenarios (train with DH, fine-tuning and testing with HH)

Table 49 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**	**Dataset size**	**AI/ML complexity**	**Horizontal Pos. accuracy at**

											CDF=90% (m)
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	DH	/	SH	25k	0	1k	4.26M*18	8.50M*18	>10
CIR	TOA	0%	DH	SH	SH	25k	0.5k	1k	4.26M*18	8.50M*18	0.28
CIR	TOA	0%	DH	SH	SH	25k	1k	1k	4.26M*18	8.50M*18	0.17
CIR	TOA	0%	DH	SH	SH	25k	2k	1k	4.26M*18	8.50M*18	0.10
CIR	TOA	0%	DH	SH	SH	25k	3k	1k	4.26M*18	8.50M*18	0.07

Table 50 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	DH	/	SH	25k	0	1k	4.26M*18	8.50M*18	13.69
CIR	TOA	0%	DH	SH	SH	25k	0.5k	1k	4.26M*18	8.50M*18	0.17
CIR	TOA	0%	DH	SH	SH	25k	1k	1k	4.26M*18	8.50M*18	0.10
CIR	TOA	0%	DH	SH	SH	25k	2k	1k	4.26M*18	8.50M*18	0.05
CIR	TOA	0%	DH	SH	SH	25k	3k	1k	4.26M*18	8.50M*18	0.04

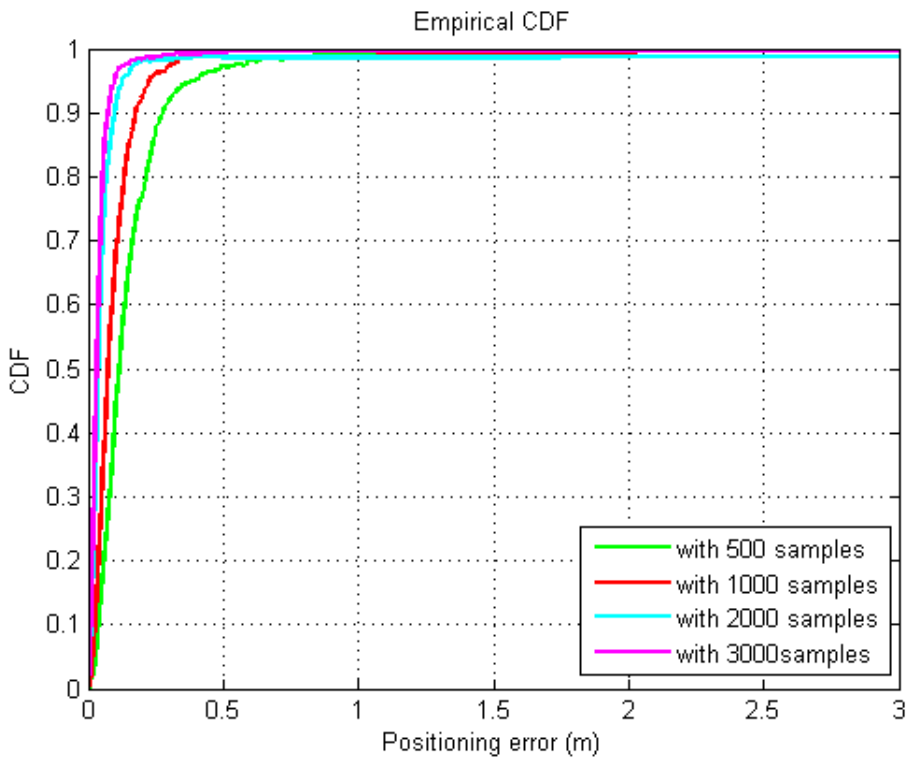


Figure 59 Positioning accuracy of model fine-tuning for different scenarios (train with DH, fine-tuning and testing with SH)

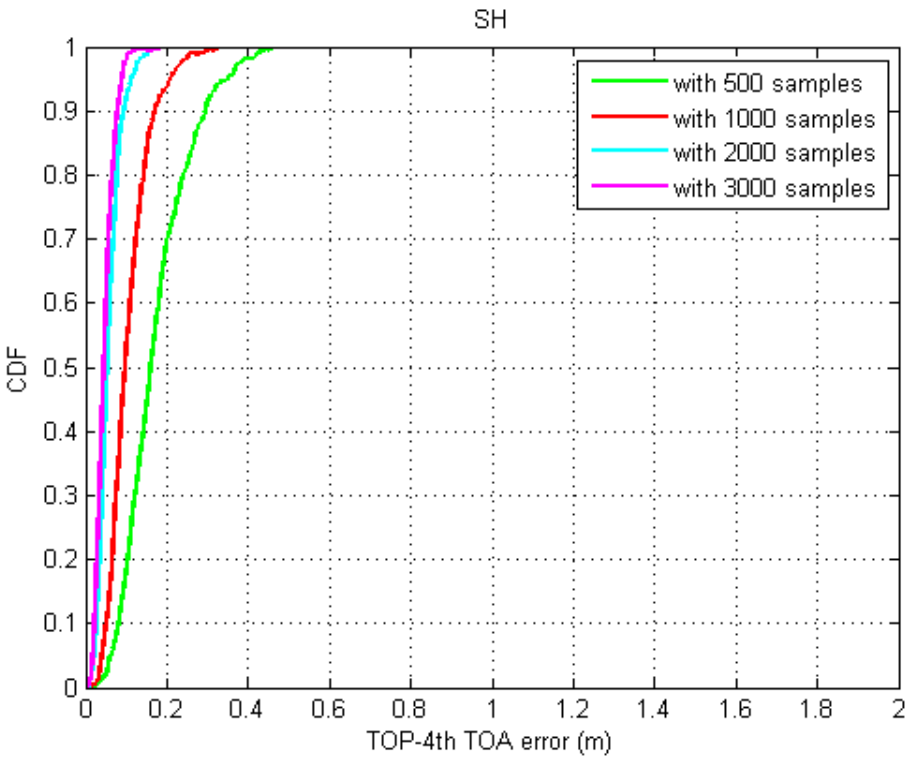


Figure 60 TOA accuracy of model fine-tuning for different scenarios (train with DH, fine-tuning and testing with SH)

Observation 35: Fine-tuning the model with small amounts of samples from an unseen scenario can achieve huge positioning accuracy improvement when the pre-trained model is transferred to such new scenario for AI/ML assisted positioning

5.2.4. Model fine-tuning across synchronization errors

We further evaluate the positioning performance when AI/ML model trained with data without synchronization error is transferred to a scenario with synchronization errors. As listed in Table 51, Table 53 and Table 55, we can observe that the synchronization-free AI/ML model would not work in these scenarios with large synchronization error (such as 50ns). However, when the pre-trained model is fine-tuned with only 1k samples of field data with 50ns synchronization error, we observe an obvious performance improvement from >8.45m@90% to 3.4m@90%. With the increase of samples used for model fine-tuning, the positioning accuracy of AI/ML model gradually improves for various synchronization errors. Moreover, the performance gain of model fine-tuning is higher for these scenarios with relatively large synchronization errors. Therefore, it is concluded that model fine-tuning can obviously mitigate the impact of synchronization errors for AI/ML assisted positioning.

Table 51 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal Pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	50ns	25k	0	1k	4.26M*18	8.50M*18	8.45
CIR	TOA	0%	0ns	50ns	50ns	25k	0.5k	1k	4.26M*18	8.50M*18	3.97
CIR	TOA	0%	0ns	50ns	50ns	25k	1k	1k	4.26M*18	8.50M*18	3.40
CIR	TOA	0%	0ns	50ns	50ns	25k	2k	1k	4.26M*18	8.50M*18	2.97

CIR	TOA	0%	0ns	50ns	50ns	25k	3k	1k	4.26M*18	8.50M*18	2.55

Table 52 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	50ns	25k	0	1k	4.26M*18	8.50M*18	6.50
CIR	TOA	0%	0ns	50ns	50ns	25k	0.5k	1k	4.26M*18	8.50M*18	3.15
CIR	TOA	0%	0ns	50ns	50ns	25k	1k	1k	4.26M*18	8.50M*18	2.83
CIR	TOA	0%	0ns	50ns	50ns	25k	2k	1k	4.26M*18	8.50M*18	2.39
CIR	TOA	0%	0ns	50ns	50ns	25k	3k	1k	4.26M*18	8.50M*18	2.38

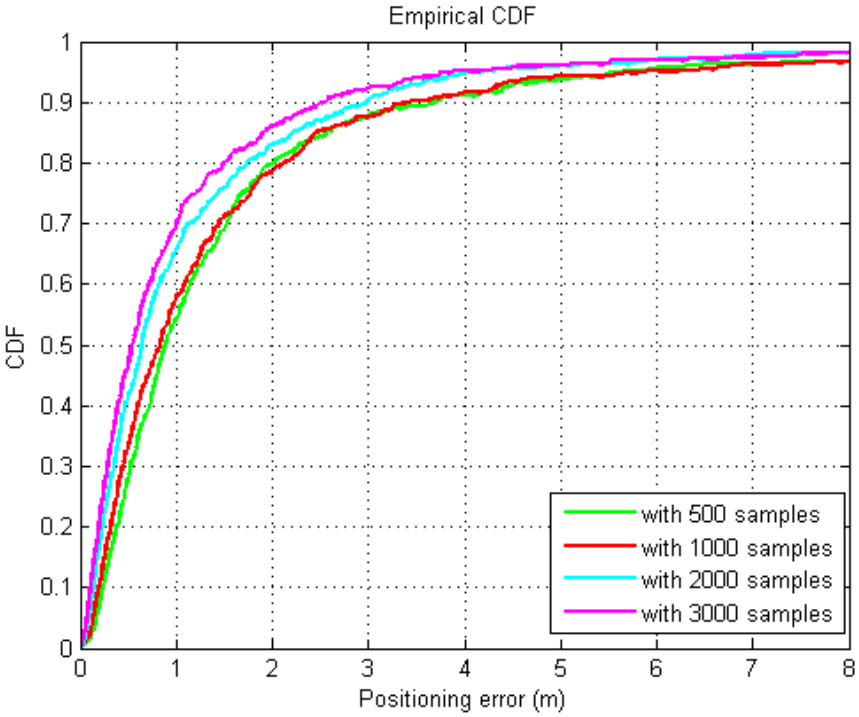


Figure 61 Positioning accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 50ns sync. error)

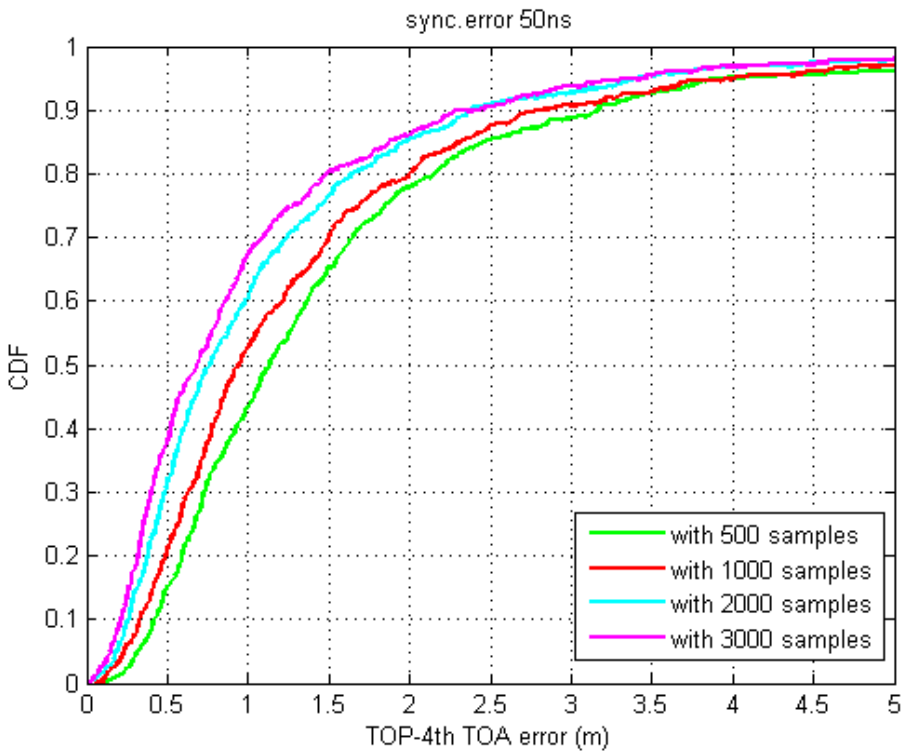


Figure 62 TOA accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 50ns sync. error)

Table 53 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal Pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	10ns	25k	0	1k	4.26M*18	8.50M*18	2.11
CIR	TOA	0%	0ns	10ns	10ns	25k	0.5k	1k	4.26M*18	8.50M*18	2.10
CIR	TOA	0%	0ns	10ns	10ns	25k	1k	1k	4.26M*18	8.50M*18	1.78
CIR	TOA	0%	0ns	10ns	10ns	25k	2k	1k	4.26M*18	8.50M*18	1.57
CIR	TOA	0%	0ns	10ns	10ns	25k	3k	1k	4.26M*18	8.50M*18	1.40

Table 54 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**TOP-4th TOA accuracy (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	10ns	25k	0	1k	4.26M*18	8.50M*18	1.76

CIR	TOA	0%	0ns	10ns	10ns	25k	0.5k	1k	4.26M*18	8.50M*18	1.76
CIR	TOA	0%	0ns	10ns	10ns	25k	1k	1k	4.26M*18	8.50M*18	1.66
CIR	TOA	0%	0ns	10ns	10ns	25k	2k	1k	4.26M*18	8.50M*18	1.50
CIR	TOA	0%	0ns	10ns	10ns	25k	3k	1k	4.26M*18	8.50M*18	1.39

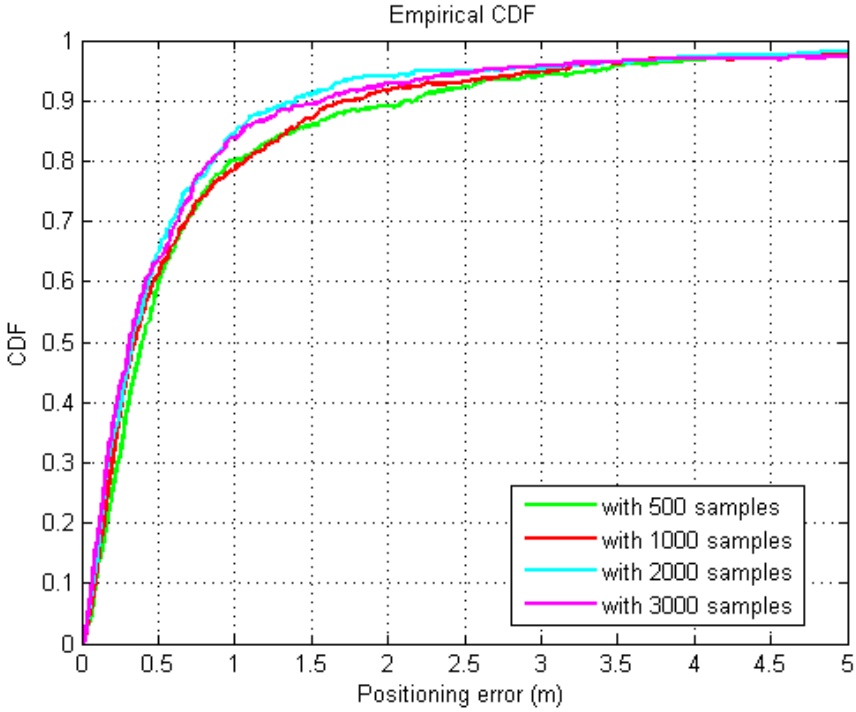


Figure 63 Positioning accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 10ns sync. error)

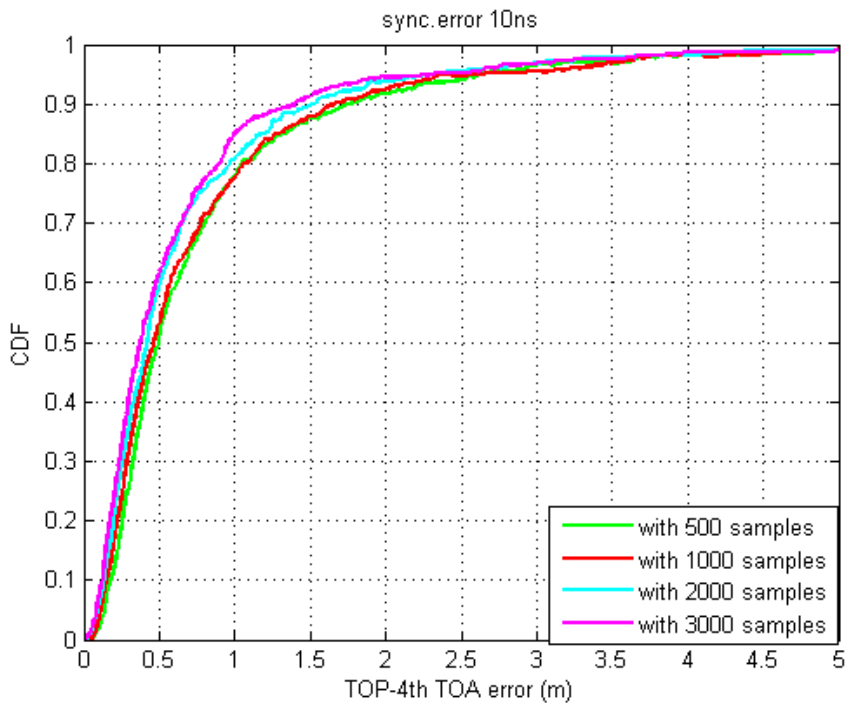


Figure 64 TOA accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 10ns sync. error)

Table 55 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)			Dataset size			AI/ML complexity		Horizontal Pos. accuracy at CDF=90% (m)
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	2ns	25k	0	1k	4.26M*18	8.50M*18	1.70
CIR	TOA	0%	0ns	2ns	2ns	25k	0.5k	1k	4.26M*18	8.50M*18	1.43
CIR	TOA	0%	0ns	2ns	2ns	25k	1k	1k	4.26M*18	8.50M*18	1.37
CIR	TOA	0%	0ns	2ns	2ns	25k	2k	1k	4.26M*18	8.50M*18	1.37
CIR	TOA	0%	0ns	2ns	2ns	25k	3k	1k	4.26M*18	8.50M*18	1.31

Table 56 Evaluation results of fine-tuning for AI/ML model deployed on UE or Network side, FNN, UE distribution area = [120x60 m]

Model input	Model output	Label	Settings (e.g., drops, clutter param, mix)			Dataset size			AI/ML complexity		TOP-4th TOA accuracy (m)
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	0%	Sync. 0ns	/	2ns	25k	0	1k	4.26M*18	8.50M*18	1.40
CIR	TOA	0%	0ns	2ns	2ns	25k	0.5k	1k	4.26M*18	8.50M*18	1.35
CIR	TOA	0%	0ns	2ns	2ns	25k	1k	1k	4.26M*18	8.50M*18	1.30
CIR	TOA	0%	0ns	2ns	2ns	25k	2k	1k	4.26M*18	8.50M*18	1.28
CIR	TOA	0%	0ns	2ns	2ns	25k	3k	1k	4.26M*18	8.50M*18	1.26

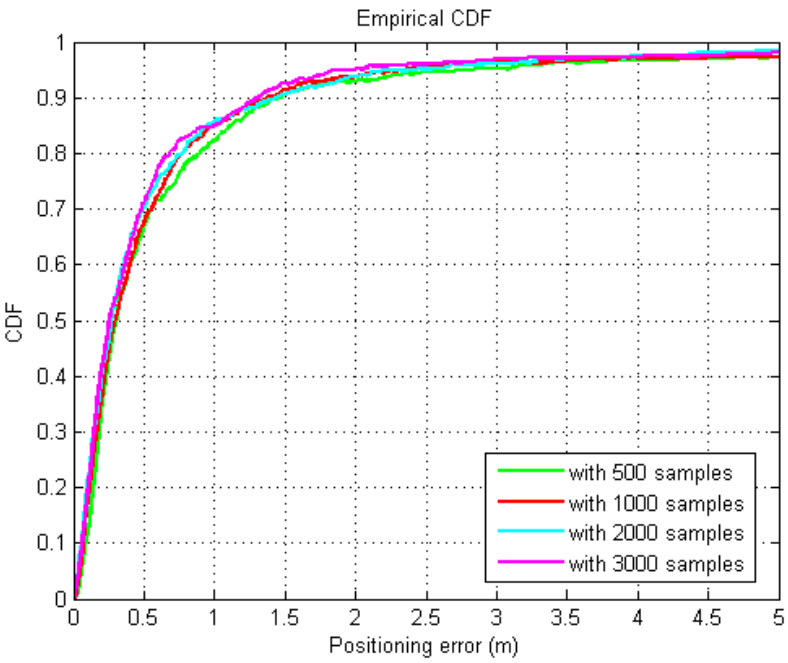


Figure 65 Positioning accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 2ns sync. error)

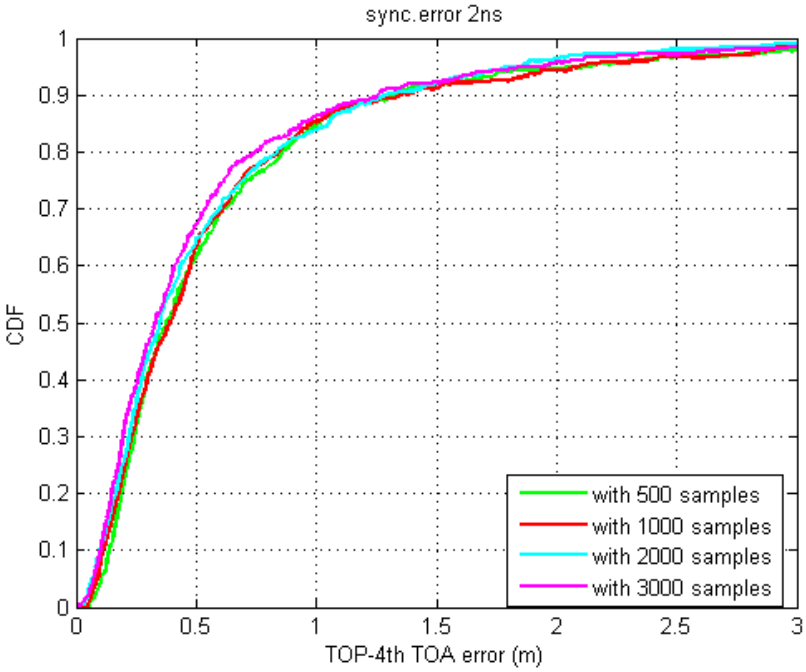


Figure 66 TOA accuracy of model fine-tuning for different synchronization errors (train without sync. error, fine-tuning and testing with 2ns sync. error)

Observation 36: Fine-tuning the model with small amounts of samples with an unseen synchronization error can achieve obvious positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such synchronization error for AI/ML assisted positioning.

Proposal 16: Further study and confirm the benefits of fine-tuning in terms of model generalization enhancement for AI/ML assisted positioning.

Proposal 17: Capture in the TR the benefits of fine-tuning for AI/ML assisted positioning in terms of positioning accuracy for AI model generalization capability.

5.3. Application of model fine-tuning

When coming back to the first question "*what scenarios or tasks are model fine-tuning applied to?*", we can find some clues from the above simulation results. According to the existing observations, it is easy to find that: the performance gain of model fine-tuning is clearly different for different cases even if fine-tuning with the same scale of field data. Furthermore, we comprehensively compare the results of different cases when 1000 samples are used for model fine-tuning. As shown in Table 57, while fine-tuning can achieve significantly performance gain, it is difficult to achieve high-accuracy positioning when there is a great difference between the source domain and the target domain, such as different scenarios for direct AI/ML positioning and different drops for AI/ML assisted positioning. There are at least two cases in which model fine-tuning with a small amount of field data can achieve high-accuracy positioning. The first case is that the target domain is greatly similar to the source domain such as different synchronization errors. The second case is that positioning in the target domain does not rely on the fingerprint feature, such as fine-tuning with SH or HH data for AI/ML assisted positioning, and in such case the target domain is easy to fit. Therefore, as for the application of model fine-tuning, we have the following observations:

Model fine-tuning is suitable for the following tasks:

- The source domain and the target domain are greatly similar, such as with different synchronization error.
- The target domain is easy to fit, such as fine-tuning with SH or HH data for AI/ML assisted positioning.

Table 57 Evaluation of model fine-tuning for different cases

Cases	Training	Fine-tuning	Testing	Positioning accuracy @90%
	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	4.40

Direct AI/ML positioning	{0.4, 2, 2}	{0.6, 6, 2}	{0.6, 6, 2}	3.23
	Drop1	Drop2	Drop2	3.97
	InF-DH	InF-HH	InF-HH	8.78
	Sync 0ns	50ns	50ns	2.39
	Sync 0ns	10ns	10ns	1.28
	Sync 0ns	2ns	2ns	1.11
AI/ML assisted positioning	{0.6, 6, 2}	{0.4, 2, 2}	{0.4, 2, 2}	0.63
	Drop1	Drop2	Drop2	5.50
	InF-DH	InF-HH	InF-HH	0.17
	InF-DH	InF-SH	InF-SH	0.17
	Sync 0ns	50ns	50ns	3.40
	Sync 0ns	10ns	10ns	1.78
	Sync 0ns	2ns	2ns	1.30

Observation 37: Model fine-tuning is suitable for the following tasks:

- **The source domain and the target domain are greatly similar, such as with different synchronization error.**
- **The target domain is easy to fit, such as TOA estimation of LOS path.**

5.4. Sample size for model fine-tuning

From the above simulation results, we find that the positioning accuracy of AI/ML model continues to improve as the increase of the field data used for model fine-tuning. Thus, it is better to collect more field data for model fine-tuning when the cost of data collection is not considered and the field data is always available. However, for a data-restricted scenario, *how many field samples are required to conduct model fine-tuning*? In order to answer this question, we further evaluate a key indicator called *data efficiency*, which means @90% positioning accuracy improvement per N additional field data. The motivation of this definition comes from the observation: with the increase of field data, the 90% positioning accuracy is improving more and more slowly and gradually tending to saturate, which means the field data becomes progressively less efficient.

5.4.1. Direct AI/ML positioning

As shown in Table 58, we present the data efficiency of different ranges of sample size (N = 100) for direct AI/ML positioning.

Table 58 Fine-tuning data sample efficiency for different cases

Cases	Range of sample size	Data efficiency (@90% per 100 additional samples)	Positioning accuracy with sample size N1 for sample range N1~N2(@90%)
Train: {0.6, 6, 2} Fine-tuning: {0.4, 2, 2} Testing: {0.4, 2, 2}	0-500	0.69	8.67 (0 samples)
	500-1000	0.16	5.22 (500 samples)
	1000-2000	0.09	4.40 (1000 samples)
	2000-3000	0.03	3.50 (2000 samples)
Train: {0.4, 2, 2} Fine-tuning: {0.6, 6, 2} Testing: {0.6, 6, 2}	0-500	0.17	4.77
	500-1000	0.13	3.89
	1000-2000	0.06	3.23
	2000-3000	0.01	2.56
Train: Drop1	0-500	0.26	6.00

Fine-tuning: Drop2 Testing: Drop2	500-1000	0.14	4.69
	1000-2000	0.06	3.97
	2000-3000	0.04	3.37
Train: DH Fine-tuning: HH Testing: HH	0-500	14.98	>>10
	500-1000	0.34	10.50
	1000-2000	0.29	8.78
	2000-3000	0.12	5.84
Train: Sync. Error 0ns Fine-tuning: 50ns Testing: 50ns	0-500	1.39	10.18
	500-1000	0.16	3.22
	1000-2000	0.06	2.39
	2000-3000	0.02	1.73
Train: Sync. Error 0ns Fine-tuning: 10ns Testing: 10ns	0-500	0.62	4.56
	500-1000	0.03	1.44
	1000-2000	0.02	1.28
	2000-3000	0.01	1.06
Train: Sync. Error 0ns Fine-tuning: 2ns Testing: 2ns	0-500	0.10	1.64
	500-1000	0.001	1.11
	1000-2000	0.01	1.11
	2000-3000	0.005	0.95

In this regard, data efficiency can be considered as a metric to determine the sample size for model fine-tuning. Figure 67 presents a curve of positioning error reduction with increasing number of sample size (per 100 additional samples). We can observe that data efficiency is very high for the first 1000 samples, and then gradually degrades with the increase of sample size. Therefore, for a data-restricted scenario, at least two methods can be exploited to determine the sample size for model fine-tuning:

- With reference to a pre-defined threshold of data efficiency, such as 0.2m/100samples (red circle);
- With reference to the saturation point of data efficiency, such as 1500 samples (black circle).

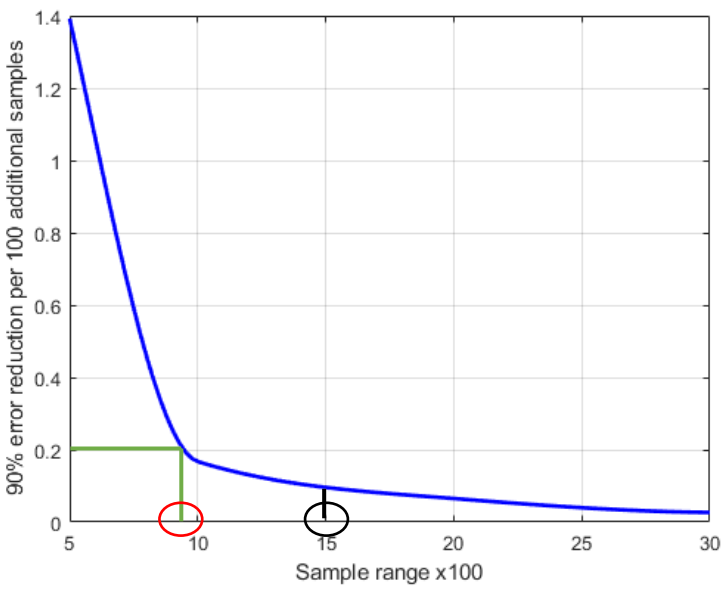


Figure 67 The curve of positioning error reduction with increasing number of sample size (data efficiency)

In general, the performance of the fine-tuned AI/ML model is positively correlated with the sample size used for model fine-tuning. Therefore, for a data-rich scenario, the minimal sample size required for model fine-tuning should depend on the target positioning performance.

5.4.2. AI/ML assisted positioning

As shown in Table 59, we present the data efficiency of different ranges of sample size (N = 100) for AI/ML assisted positioning.

Table 59 Fine-tuning data sample efficiency for different cases

Cases	Range of sample size	Data efficiency (@90% per 100 additional samples)	Positioning accuracy with sample size N1 for sample range N1~N2(@90%)
Train: {0.6, 6, 2} Fine-tuning: {0.4, 2, 2} Testing: {0.4, 2, 2}	0-500	0.570	3.70 (0 samples)
	500-1000	0.044	0.85(500 samples)
	1000-2000	0.015	0.63(1000 samples)
	2000-3000	0.001	0.48(2000 samples)
Train: Drop1 Fine-tuning: Drop2 Testing: Drop2	0-500	0.952	10.37
	500-1000	0.022	5.61
	1000-2000	0.047	5.50
	2000-3000	0.095	5.03
Train: DH Fine-tuning: HH Testing: HH	0-500	1.99	20.20
	500-1000	0.026	0.30
	1000-2000	0.008	0.17
	2000-3000	0.003	0.09
Train: DH Fine-tuning: SH Testing: SH	0-500	2.042	20.70
	500-1000	0.022	0.28
	1000-2000	0.007	0.17
	2000-3000	0.001	0.10
Train: Sync. Error 0ns Fine-tuning: 50ns Testing: 50ns	0-500	0.896	8.45
	500-1000	0.114	3.97
	1000-2000	0.043	3.40
	2000-3000	0.042	2.97
Train: Sync. Error 0ns Fine-tuning: 10ns Testing: 10ns	0-500	0.002	2.11
	500-1000	0.064	2.10
	1000-2000	0.021	1.78
	2000-3000	0.017	1.57
Train: Sync. Error 0ns Fine-tuning: 2ns Testing: 2ns	0-500	0.054	1.70
	500-1000	0.012	1.43
	1000-2000	0.001	1.37
	2000-3000	0.006	1.37

Proposal 18: Both data efficiency and target performance could be considered as reference to determine the sample size required for model fine-tuning.

5.5. Discussion on model fine-tuning/retraining and mix-training

As evaluated early, both of model fine-tuning/retraining and mix-training can obviously improve model generalization when the well-trained AI/ML model is transferred to other scenarios. In general, the main difference is that mix-training requires that the data of all deployed scenarios has been sampled or constructed in advance, and thus the AI/ML model trained with the mixed dataset can naturally adapt to these scenarios. Model fine-tuning is to fine-tune/retrain the pretrained AI/ML model with the newly collected field data, which is more flexible without strict requirement on the data for model pretraining. However, model fine-tuning may also face some challenges, such as catastrophic forgetting when only a small amount of training data is used to fine-tune a large pre-trained model, heavy reliance on dataset size for these complicated changes (such as different drops).

In summary, model fine-tuning can flexibly adapt AI/ML model to various dynamic changes in environment, while mix-training is more suitable for these static changes. It is benefit to integrate model fine-tuning and mix-training so as to fully leverage their advantages. For example, the dataset used for model fine-tuning contains not only the field data, but also the data from original training data used for pretraining the AI/ML model.

Proposal 19: Model fine-tuning can flexibly adapt AI/ML model to various dynamic changes in environment, while mix-training is more suitable for these static changes.

Proposal 20: It is benefit to integrate model fine-tuning and mix-training so as to fully leverage their advantages.

6. Model training framework with fewer labeled data

At the RAN1#110 meeting, it was agreed that：

Agreement

For evaluation of AI/ML based positioning, study the performance impact from availability of the ground truth labels (i.e., some training data may not have ground truth labels). The learning algorithm (e.g., supervised learning, semi-supervised learning, unsupervised learning) is reported by participating companies.

For supervised learning, large-scale and high-quality training data is of great importance for model performance. However, it may be difficult to collect large-scale training data for AI/ML based positioning, especially accurate location labels. There are three possible solutions to train an AI/ML model with fewer labeled data, including fine-tuning, semi-supervised learning, and multiple antenna ports. Specifically,

- Fine-tuning can achieve very good positioning accuracy while requiring a well-trained AI model in advance, and how to obtain this model is also an open issue.
- Semi-supervised learning can improve positioning accuracy with the assistance of some extra unlabeled data, where the unlabeled data is relatively easy to collect.
- Multi-port data can also be utilized to improve the positioning accuracy, while more ports resource may be required to support data collection and measurement.

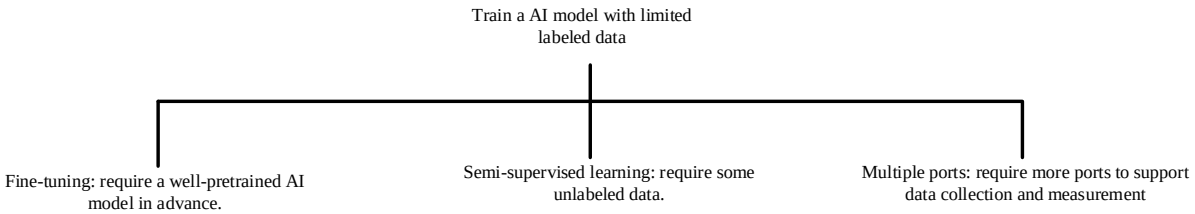


Figure 68 Three possible solutions to train an AI/ML model with limited labeled data.

Considering that high-precision positioning is an essential technology to empower intelligence for future applications, such as smart cities and smart factories, AI/ML-based High-Precision Positioning is selected as a Track for the 3rd Wireless Communication AI Competition (WAIC) which is organized by CAICT, vivo and Huawei [15]. In particular, how to jointly utilize fewer labeled data and large-scale unlabeled data to improve positioning accuracy is adopted as one of three scenarios. According to the statistics after the competition, the players' schemes can achieve a positioning accuracy of less than 1m@90% when only 1k labeled samples are provided, as shown in Table 60. Specifically, their schemes mainly consist of data augmentation to labeled data, semi-supervised learning (Pseudo-label based and Contrastive learning based) and fine-tuning. Therefore, we strongly believe that the study on model training framework with fewer labeled data has great potential to achieve high-precision positioning with lower cost of data collection from the perspective of model implementation in future.

Table 60 Players' Ranking for WAIC

Total Ranking	Positioning error (@90%) (Scenario 3)
1	0.39m
2	0.40m
3	0.71m
4	0.75m
5	0.75m
6	0.80m
7	1.01m
8	0.90m
9	1.00m
10	1.10m

6.1. Model fine-tuning with limited filed data

As presented in section 4.4.4, we mainly evaluate and analyze the performance of model fine-tuning from the perspective of model generalization. Moreover, model fine-tuning is also an effective way to train a scenario-specific AI model quickly with less field data requirement for a new scenario. From the simulation results in section 4.4.4, we observe that fine-tuning the pretrained model with only 1k collected field samples yields significant performance gain. Meanwhile, fewer computation and storage resources are required to train such a new model as compared to large-scale model training from scratch. In this sense, both of filed data collection and model fine-tuning can also be conducted at UE side as well. However, the performance of model fine-tuning relies on a well-pretrained model, and how to obtain this model is still an open issue.

6.2. Semi-supervised learning with limited labeled data

AI/ML is data-driven, and the excellent performance benefits from a large number of available training data. In practice, some labeled data can be collected by Positioning Reference Unit (PRUs) deployed in a network. However, it is difficult to collect enough labeled data to enable large-scale model training for the use case of AI/ML based positioning accuracy enhancement, which motivates us to investigate the AI/ML technologies with low labeled data dependence. Fortunately, the unlabeled data containing CIR only is relatively easy to obtain. For example, one way to collect unlabeled data at network side is that UEs report CIRs estimated from PRS measurement. Given that we have large amounts of data without location labels but relatively small amounts of data with location labels, we hope to train a high-accuracy AI/ML model with these data. Semi-supervised learning may be also an effective way to tackle this challenging task.

In essence, AI/ML model inference mainly utilizes three features of CIR, including first path information due to the existence of absolute time of arrival, fingerprint information due to the existence of spatial consistency, and correlation of CIRs for fixed TRPs' topology. Among them, we observe that the positioning accuracy of spatial consistency settings is greatly better than that of non-spatial consistency settings, and thus the fingerprint information is significantly important for positioning. Moreover, for traditional supervised learning, the fingerprint information can be captured by AI/ML models only when there are large amounts of labeled training data. In other words, when there are only small amounts of labeled data, the fingerprint information can not be completely extracted and utilized by AI/ML models. In this context, we resort to semi-supervised learning to capture the fingerprint information from both labeled data and unlabeled data. Specifically, we propose an iterative semi-supervised learning framework by integrating the advantages of channel charting, fine-tuning and contrastive learning.

- Channel charting [6]: map the high-dimension CSI to a low-dimension manifold space following neighbor reservation.
- Fine-tuning [7]: adjust the model with labeled field data.
- Contrastive learning [8]: a kind of self-supervised learning, that is, learning differences from dissimilar samples and learning similarities from similar samples without reliance on labeled data.

The simulation results are listed in Table 61. We can observe that semi-supervised learning can significantly improve positioning accuracy by utilizing limited labeled data and a large number of unlabeled data.

Table 61 Evaluation results of semi-supervised learning for AI/ML model deployed on UE or Network side, without model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Clutter param**	**Dataset size & type**		**AI/ML complexity**		**Horizontal positioning accuracy at CDF=90% (meters)**
				Train	test	Model complexity	Computational complexity	AI/ML
CIR	Pos.	96%	{0.6, 6, 2}	1k labeled & 25k unlabeled	1k	1.65M	22.30M	5.05
CIR	Pos.	99%	{0.6, 6, 2}	0.3k labeled & 25k unlabeled	1k	1.65M	22.30M	8.78
CIR	Pos.	0	{0.6, 6, 2}	1k	1k	1.65M	22.30M	12.06
CIR	Pos.	0	{0.6, 6, 2}	2k	1k	1.65M	22.30M	9.03
CIR	Pos.	0	{0.6, 6, 2}	2k	1k	1.65M	22.30M	5.53

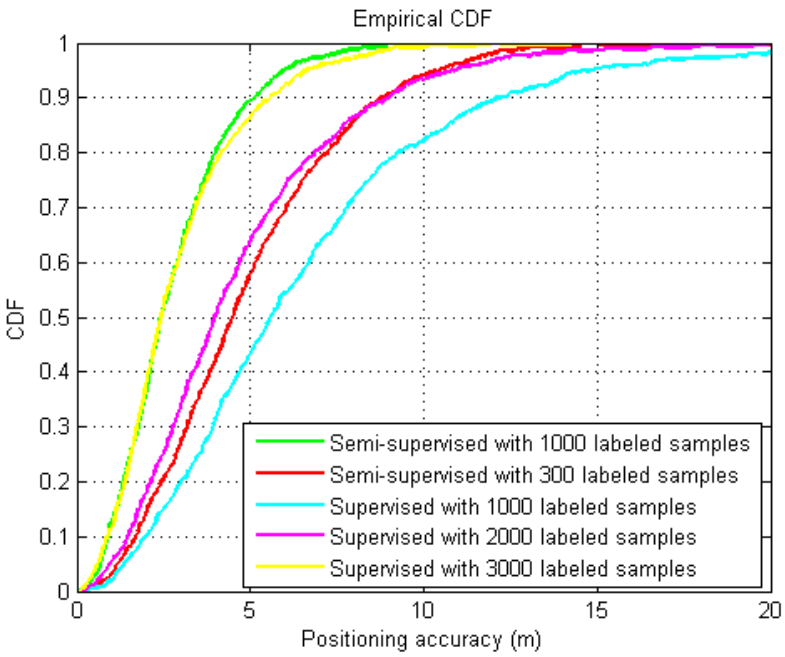


Figure 69 Positioning accuracy comparison of semi-supervised learning and supervised learning with different numbers of labeled samples

Observation 38: Semi-supervised learning can achieve a more accurate position estimation as compared to supervised learning with less amount of labeled data.

Proposal 21: Capture in the TR the benefits of semi-supervised learning for AI/ML based positioning in terms of less data collection for training and more positioning accuracy.

6.3. Positioning with multiple ports data

There are two types of errors for AI/ML inference, i.e., bias and variance. Bias is caused by the model's inability to represent current data distribution, such as an AI/ML model trained with data distribution A but tested with data distribution B, which can be solved by transfer learning–like methods and retraining. Variance is caused by the imperfection of the model and data. Specifically, overfitting is everywhere, resulting in that the AI/ML model can only find a 'local' law but never find the 'global' law due to limited training data sampled from the physical world. Moreover, data measurement may be subject to fluctuations. For example, even at the same location, the channels measured at different times by different terminals can be different. In this sense, this imperfection may result in the fluctuation of predicted results of the AI/ML model around the true labels.

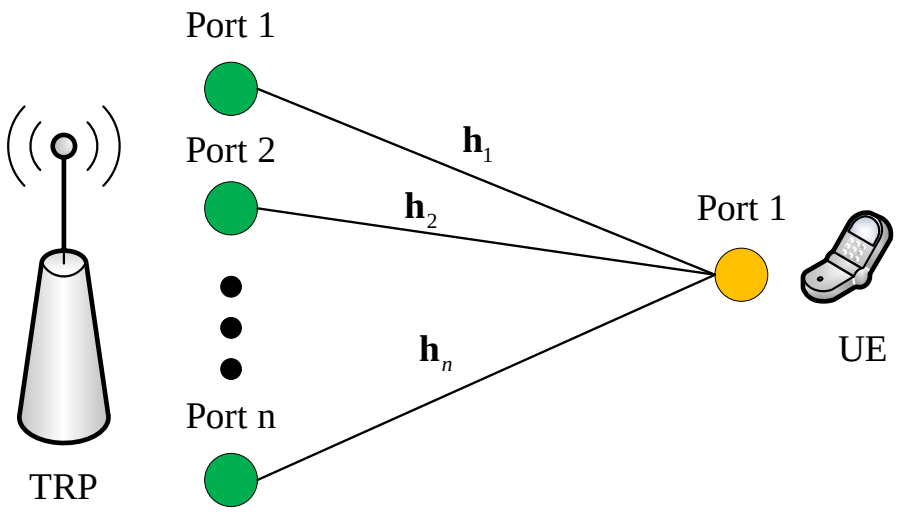


Figure 70 A scenario of multi-port positioning.

In this context, we try to resort to multiple port data to reduce the variance of AI/ML model inference, and further the positioning accuracy can be improved. Specifically, assuming that there are n PRS ports at each TRP and 1 PRS port at UE, we can divide T training samples with shape $(T\times 256\times 18\times n\times 1)$ into nT training samples with shape $(nT\times 256\times 18)$. In this way, the scale of training dataset is increased, and AI/ML model can be trained with this scaled dataset. At each model inference, the CIRs from n ports are separately estimated and then fed into the AI/ML model. Then, AI/ML model will output n positioning results corresponding to n ports' input. Finally, a more accurate position estimation can be obtained by fusing n positioning results, especially when some prior knowledge about each port is known, such as channel quality and testing error of each port. When the prior testing error of each port is available, a possible fusion method is described as follows.

$$\mathbf{p}^* = \sum_{i=1}^{n} \frac{e^{-a_i}}{\sum_{t=1}^{n} e^{-a_t}} \mathbf{p}_i$$

where a_i denotes the prior testing error of i-th port. Moreover, a simple linear average method can also be adopted when there is no prior knowledge of each port.

$$\mathbf{p}^* = \frac{1}{n}\sum_{i=1}^{n} \mathbf{p}_i$$

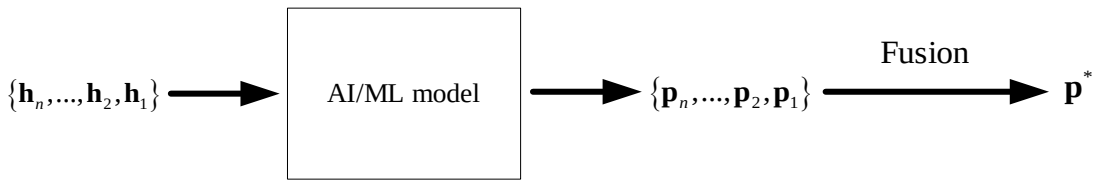


Figure 71 A positioning framework with multi-port data.

Another possible method is to train AI/ML model with multi-port data directly without division, i.e., CIR with shape $(256\times 18\times n\times 1)$ as the input, but this model can work only when n-port CIR is always available.

We assume that there are 8 ports at each TRP and 1 port at UE and 3k samples are used to train the AI/ML model. As shown in Table 62, the simulation results indicate that multi-port positioning can achieve higher positioning accuracy as compared to single-port positioning at the cost of more resource requirements for PRS transmission and measurement. Note that each training sample corresponds to a UE.

Table 62 Evaluation results of multiple ports for AI/ML model deployed on UE or Network side, without model generalization, ViT, UE distribution area = [120x60 m]

Model input	**Model output**	**Label**	**Clutter param**	**Dataset size & type**		**AI/ML complexity**		**Horizontal positioning accuracy at CDF=90% (meters)**
				Training	test	Model complexity	Computational complexity	AI/ML
CIR	Pos.	0	{0.6, 6, 2}	3k & 8 ports	1k & 8 ports	1.65M	22.30M	3.14
CIR	Pos.	0	{0.6, 6, 2}	3k & 1 port	1k & 1 port	1.65M	22.30M	5.53

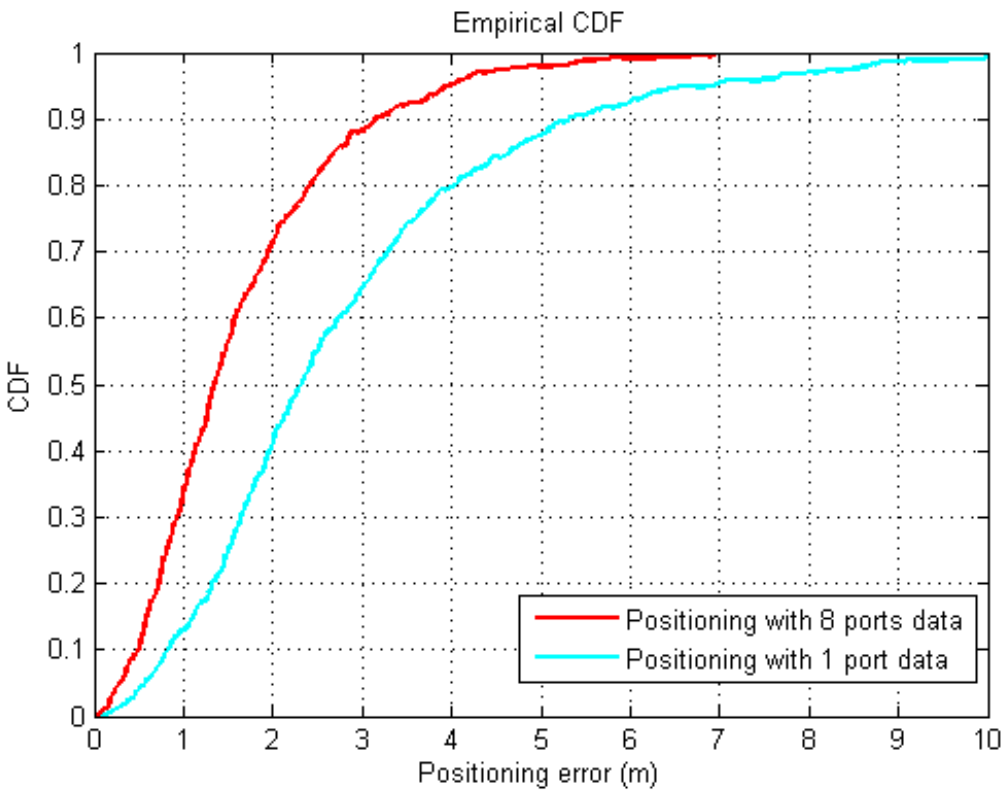


Figure 72 CDF of positioning accuracy of multi-port positioning and single-port positioning.

Observation 39: Positioning with multi-port data can achieve a more accurate position estimation as compared to single-port positioning.

Proposal 22: Capture in the TR the benefits of multi-port positioning for AI/ML based positioning in terms of positioning accuracy.

7. Evaluation for model monitoring

On the agenda 9.2.1 of RAN1 #110bis-e meeting, it was agreed that

Agreement

Study at least the following metrics/methods for AI/ML model monitoring in lifecycle management per use case:

- Monitoring based on inference accuracy, including metrics related to intermediate KPIs
- Monitoring based on system performance, including metrics related to system performance KPIs
- Other monitoring solutions, at least following 2 options.
  0. Monitoring based on data distribution
     a) Input-based: e.g., Monitoring the validity of the AI/ML input, e.g., out-of-distribution detection, drift detection of input data, or ~~something simple like checking~~ SNR, delay spread, etc.
     b) Output-based: e.g., drift detection of output data
  1. Monitoring based on applicable condition

Note: Model monitoring metric calculation may be done at NW or UE

At RAN1 #111 meeting, it was agreed that

Agreement

For AI/ML assisted approach, study the performance of model monitoring metrics at least where the metrics are obtained from inference accuracy of model output.

Agreement

- Regarding AI/ML model monitoring for AI/ML based positioning, to study and provide inputs on feasibility, potential benefits (if any) and potential specification impact at least for the following aspects
  - At least the following are identified for further study as potential data for calculating monitoring metric
    - If monitoring based on model output
      - E.g. , estimated UE location corresponding to model output for direct AI/ML positioning, estimated intermediate parameter(s) corresponding to model output for AI/ML assisted positioning, ground truth label corresponding to model inference output for both direct and AI/ML assisted positioning
    - If monitoring based on model input
      - E.g., measurement corresponding to model inference input
    - Note1: other type of potential data for model monitoring is not precluded
    - Note2: combination of one or more type of potential data for monitoring is not precluded
  - If a given type of data is necessary for calculating monitoring metric, study whether and if so
    - How an entity can be used to provide the given type of data for calculating monitoring metric
      - Companies are requested to report their assumption of the entity (or entities) used to provide the given type of data for calculating monitoring metric for each case
    - Potential signalling for provisioning of the given type of data for calculating associated monitoring metric
    - Potential assistance signaling and procedure to facilitate an entity providing data for calculating monitoring metric
    - Potential UE-network interaction
      - E.g., model monitoring decision indication between UE and network

At RAN1 #112 meeting, it was agreed that

Agreement

For **direct** AI/ML positioning, study the performance of model monitoring methods, including:

- Label based methods, where ground truth label (or its approximation) is provided for monitoring the accuracy of model output.
- Label-free methods, where model monitoring does not require ground truth label (or its approximation).

Model monitoring is a key component of lifecycle management (LCM), and play an essential role in ensuring positioning accuracy for AI/ML based positioning. Moreover, model monitoring may be also related to other LCM procedures, such as data collection, model updating and model inference. Regarding its complexity and importance, it is necessary to build the specific simulation method to evaluate emerging model monitoring schemes, such as for model input and output based model monitoring as agreed in previous meetings. In this section, we present our model monitoring schemes, including specific simulation assumption, scheme design and simulation results. Moreover, we present our understanding on the basic principle of model monitoring from the perspective of dataset shift and Bayesian theory in our companion contribution [14].

At RAN1 #112 meeting, model monitoring methods are divided into two categories, i.e., label based methods and label-free methods. For the following model monitoring methods, only section 7.2.1 *Ground truth label based model monitoring* is the label based method, and the others are label-free methods.

7.1. Model monitoring based on model input

Model invalidation is mainly caused by the non-stationary environment, and these changes of environment may shift the distribution of model input, e.g., CIR. From this view, detecting the distribution shift of model input is a feasible method for model monitoring. For convenience, we only evaluate the performance of model monitoring for cases of adopting CIR as model input, and other cases of adopting other types of measurement such as RSRP is also theoretically feasible. Moreover, considering the availability of ground truth labels, model input based model monitoring without need of ground truth labels is more accessible at least for the use case of AI/ML based positioning.

7.1.1. The shift detection of dominant feature distribution

CIR can be characterized by various dominant features, such as multi-path delay, RSRP, SINR, delay spread and so on. The distribution of CIR can be regarded as a combination of that of these dominant features, and the distribution shift of any features may cause the distribution shift of CIR. In practice, these dominant features are relatively easy to measure or estimate from known RS or CIR. Thus, monitoring the distribution of dominant feature may be a more efficient method as compared with directly monitoring the distribution shift of high-dimension CIR vectors.

We present an example when adopting SINR as a dominant feature for model monitoring. As shown in Figure 30 and Figure 31, the distribution shift of SINR results in obvious performance degradation. Therefore, at least adopting SINR as a dominant feature of CIR is valid for model monitoring. It is worth nothing that not all dominant features contribute to positioning, and it is possible that the distributions of some dominant features shift but the positioning accuracy still holds. Thus, identifying these dominant features strongly related to positioning is meaningful for model monitoring.

How to measure the distance between two distributions? Mathematically, there are many metrics that can describe the difference between two distribution, such as *maximal vertical distance* between two CDF curves as shown in Figure 73, *cross entropy, KL divergence* and so on, which can be reused directly here. As presented in Table 63, when taking the SINR distribution without interference as the reference distribution and other cases as the test distributions, it is observed that the metrics of both maximal vertical distance and KL divergence increase with the deviation of the test distribution from the reference distribution increases.

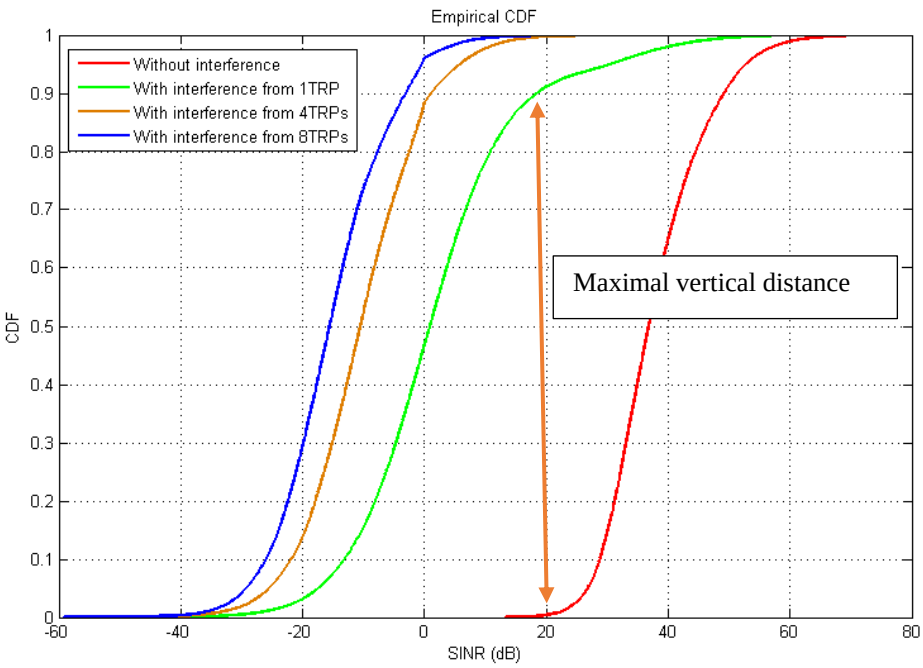


Figure 73 Illustration of maximal vertical distance

Table 63 Evaluation of distance metrics between SINR distributions

Distribution for reference	Distribution for test	Maximal vertical distance	KL divergence
Without interference	With inference from 1TRP	0.88	3.96

Without interference	With inference from 4TRP	0.99	34.70
Without interference	With inference from 8TRP	0.99	41.20

Observation 40: Adopting SINR as a dominant feature of CIR is valid for model monitoring

Proposal 23: When model input is CIR or PDP, identify these dominant features strongly related to positioning for model monitoring

Proposal 24: The metrics that can describe the difference between two distributions mathematically can be reused directly for model monitoring.

7.1.2. AI/ML based adversarial validation

Apart from the dominant features, CIR also consists of other latent features which are closely related to positioning. Model monitoring purely relying on dominant features may be partial, and can result in a biased model monitoring result. Therefore, it is better to utilize all features of CIR to perform model monitoring. Luckily, AI/ML technology can be utilized to extract all these latent and dominant features from a high-dimension CIR vector.

The main idea of adversarial validation is to construct a classifier to classify original training dataset and test dataset. Here, the original training dataset refers to the dataset used to train the AI/ML model for positioning, and the test dataset refers to the dataset collected from the real environment. If the classifier can not distinguish the original training dataset and test dataset, it means that the distributions of the two datasets are same. If the classifier can clearly distinguish the original training dataset and test dataset, it means that the distributions of original training dataset and test dataset are different, which is called *out of distribution*. The AI/ML model may suffer from severe performance degradation when out of distribution occurs. As shown in Figure 74, the specific implementation is disclosed as follows:

- **Constructing training dataset for classifier**: the training dataset is composed of two parts. The first is the original training dataset used to train the AI/ML model for positioning. The second is the dataset collected from the real environment, which is used to monitor the performance of the AI/ML model. Initially, assume that these two datasets stem from different distributions. All samples within the original training dataset are labeled as category '1', while all samples within the test dataset are labeled as category '0'.
- **Model training**: train a binary classification model with the constructed training dataset.
- **Cross validation**: validate the classification accuracy of the trained binary classification model. If the classification accuracy is larger than a predefined threshold (e.g., 90%), it means the distributions of original training dataset and test dataset are significantly different, and thus the AI/ML model for positioning suffers from performance degradation. If the classification accuracy is smaller than a predefined threshold, it means the distributions of the two datasets are generally same, and thus the performance of AI/ML model for positioning still holds.

As for simulation assumption, the original training dataset is composed of 10000 samples of drop1 DH{0.6, 6, 2}, and the AI/ML model to be monitored is trained with 25000 samples of drop1 DH{0.6, 6, 2}. The test dataset is composed of 1000 samples of drop2 DH{0.6, 6, 2}, which denotes the samples collected from the real environment. When the classifier is well-trained, another dataset consisting of 3000 samples of drop1 and 3000 samples of drop2, is used to validate the classification accuracy. Figure 75 illustrates that the classifier can distinguish the samples accurately (AUC = 0.999) by cross validation when the original training dataset and test dataset comes from different drops, which means that the current AI/ML model is invalid or suffers from severe performance degradation. As a comparison, when the original training dataset and test dataset come from the same drop, the classifier can not distinguish the samples accurately (AUC = 0.497) by cross validation, which means the current AI/ML model is valid.

The main drawback of adversarial validation is the requirement of on-device training, which may limit the accessibility of these terminals without mode training capabilities. One solution is to transfer the collected test data to another OTT server or network entity, and then model training of such classifier can be performed on these entities. In this way, adversarial validation can achieve model monitoring at the cost of acceptable hardware resource consumption for model training.

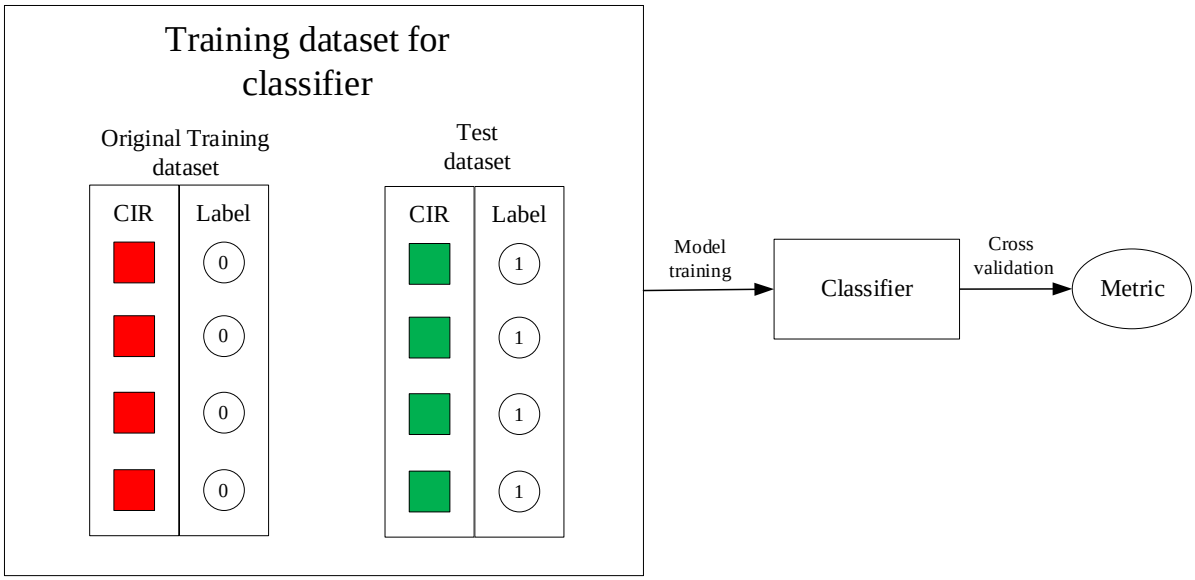


Figure 74 Illustration of adversarial validation

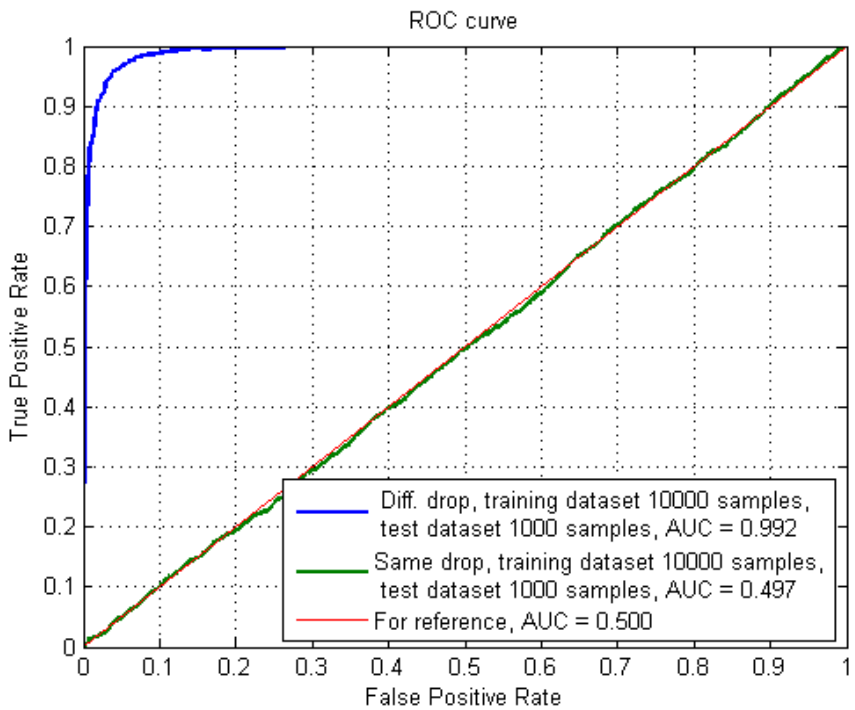


Figure 75 Result of adversarial validation based model monitoring

Observation 41: The proposed adversarial validation can achieve accurate model monitoring at the cost of acceptable hardware resource consumption for model training.

7.1.3. AI/ML based out-of-distribution detection

Thanks for the powerful capabilities of feature extraction, we have verified that AI/ML technology can be used not only for positioning function, but also for model monitoring in Section 7.1.2. To avoid frequent model training, we further propose an offline training based model monitoring scheme named *AI/ML based out-of-distribution detection*. The main idea behind is that an AI/ML model can be utilized to learn the difference between the original training dataset and non-original training dataset. For model inference, when a distribution-unknown CIR is input into the well-trained AI/ML model, the AI/ML model can indicate whether this CIR belongs to the distribution of the original training dataset or the likelihood belonging to the distribution of the original training set. As shown in Figure 76, how to implement this scheme is disclosed as follows:

- **Constructing training dataset for classifier:** the training dataset is composed of two parts. The first is the original training dataset used to train the AI/ML model for positioning. The second is the non-original training dataset whose distribution should be different from that of original training dataset. Specifically, the non-

original training dataset can be collected from **other** real environments and even generated by simulation. Then, all samples within the original training dataset are labeled as category '1', while all samples within the non-original training dataset are labeled as category '0'.

- **Model training:** train a binary classification model offline with the constructed training dataset. In this way, the bound of the original training dataset distribution can be learned.
- **Model inference**: when a distribution-unknown CIR collected from the real environment is input into the well-trained AI/ML model, the AI/ML model can make a prediction on whether this CIR belongs to the distribution of the original training dataset or the likelihood belonging to the distribution of the original training set. When the proportion of samples belonging to the distribution of the original training dataset is larger than a predefined threshold (e.g., 90%), it means that the current AI/ML model for positioning is valid. Otherwise, the current AI/ML model for positioning is invalid.

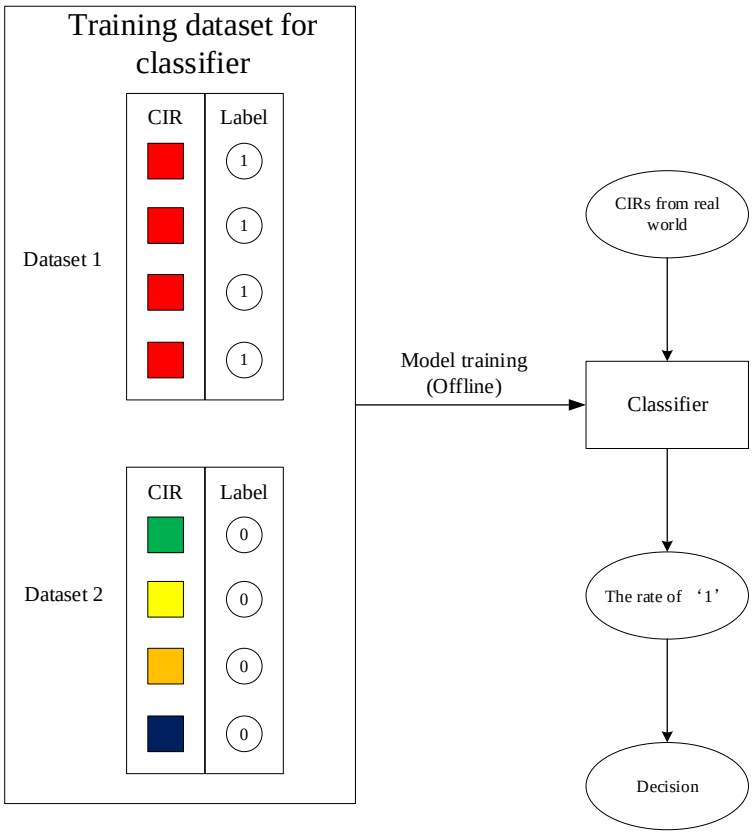


Figure 76 Illustration of AI/ML based out-of-distribution detection

As for simulation assumption, the original training dataset is composed of 10000 samples of drop1 DH{0.6, 6, 2}, and the non-original training is composed of 5000 samples of drop2 DH{0.6, 6, 2}and 5000 samples of HH. Moreover, 3000 samples of drop1 DH{0.6, 6, 2} and 3000 samples of SH{0.2, 2, 10} are utilized to test the accuracy of the binary classification model for model monitoring. As shown in Figure 77, it is observed that the classifier can distinguish the samples accurately (AUC = 0.999) whether for samples from the training set or samples from the non-original training dataset. Thus, it is concluded that AI/ML based out-of-distribution detection can achieve flexible and accurate model monitoring without need of frequent model training and large-scale data collection.

Observation 42: The proposed AI/ML based out-of-distribution detection can achieve accurate and flexible model monitoring without efforts of model training and large-scale data collection.

Proposal 25: Further study model input based model monitoring schemes.

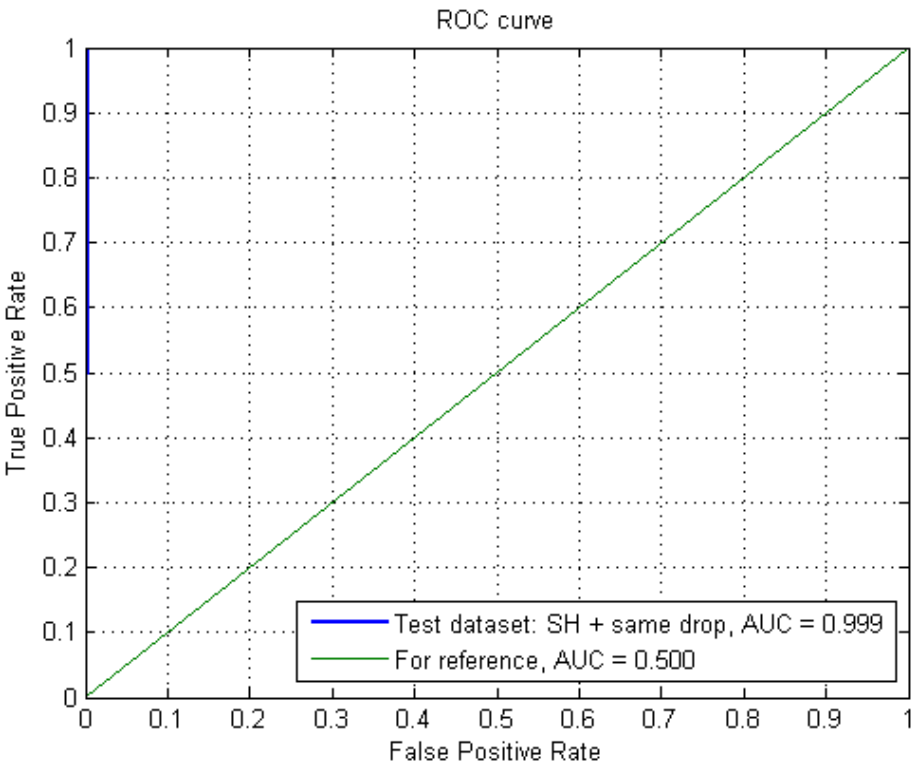


Figure 77 Result of AI/ML based out-of-distribution detection

7.2. Model monitoring based on model output

For model output based monitoring, there are two possible ideas. The first is to monitoring the distribution of model output, e.g., UE's location. When the distribution of estimated locations of AI/ML model is out of the original distribution, it means that the current AI/ML model for positioning suffers from performance degradation. Another more general idea is to monitoring the mapping relationship between model input space and output space. Once the environment changes, this mapping relationship is expected to change accordingly.

Since the AI/ML model for positioning is a discriminative model essentially, model output based model monitoring seems more intuitive than its counterpart. *However, the performance of model monitoring highly relies on the accuracy of reference information used for model monitoring, and at least it should be higher than the positioning accuracy of the current AI/ML model*. For example, the accuracy of ground truth label (in Section 7.2.1) and MSI measurement (in Section 7.2.2) should be higher than the positioning accuracy of the current AI/ML model. Only in this way can we have enough confidence to judge that model invalidation is caused by positioning error of AI/ML model but not other factors, when model monitoring metric deviates from the theoretical value.

7.2.1. Ground truth label based model monitoring

When ground truth labels can be collected from deployed PRUs, comparing the difference between the location estimated by AI/ML model and the corresponding ground truth label is the most direct and reliable manner to monitor the mapping relationship between model input space and output space. However, obtaining such extensive ground truth labels may be difficult due to limited number of PRUs deployed in the network, and model monitoring based on the positioning performance of several PRUs within a broad area may be reckless. To deal with this issue, a possible solution is to adopt the positioning results of other positioning methods as ground truth labels, but the positioning accuracy of other positioning methods should be higher than that of the AI/ML model as mentioned earlier, which is different to guarantee in practice.

Proposal 26: The accuracy and quantity of ground truth labels should be considered for ground truth label based model monitoring

7.2.2. Motion sensors assisted model monitoring

With the rapid development of sensor technology, terminals may deploy multiple high-accuracy sensors with distinctive functions at a low hardware cost. In the past few decades, motion sensors such as accelerometers, gyros, magnetometers, have been widely deployed at various type of mobile terminals to calculate the motion state information (MSI) of UE. Primarily, MSI contains the velocity v and direction θ of motion. According to the measurement results [11], motion sensors can achieve a high-accuracy measurement with errors less than

$\Delta v = 5cm/s$ and $\Delta\theta = 0.4\deg/s$. Beneficial from its high reliability and widely deployment, combining the measurement of motion sensors and legacy positioning methods to improve the positioning accuracy has been supported by Release 17 [13]. Here, we expect that motion sensors can be utilized to assist model monitoring for AI/ML based positioning when assuming that high-precision MSI measurements can be obtain by some terminals.

As illustrated in Figure 78, the specific procedure of MSI assisted model monitoring are described as follows:

- At time slot t, the terminal obtains its location estimation $\hat{\mathbf{p}}_t$ by AI/ML model, and records its MSI (v_t, θ_t).
- At time slot $t+\Delta t$, the terminal obtains its location estimation $\hat{\mathbf{p}}_{t+\Delta t}$ by AI/ML model, and records its MSI $(v_{t+\Delta t}, \theta_{t+\Delta t})$.
- The terminal predicts its location $\widehat{\mathbf{p}}_{t+\Delta t}$ at time slot $t+\Delta t$ based on the location estimation $\hat{\mathbf{p}}_t$ and the measured MSI (v_t, θ_t).
- Calculate the Euclidean distance d between $\widehat{\mathbf{p}}_{t+\Delta t}$ and $\hat{\mathbf{p}}_{t+\Delta t}$. When the distribution of d exceeds a predefined threshold, it means the current AI/ML model is invalid. Otherwise, the current AI/ML model still works.

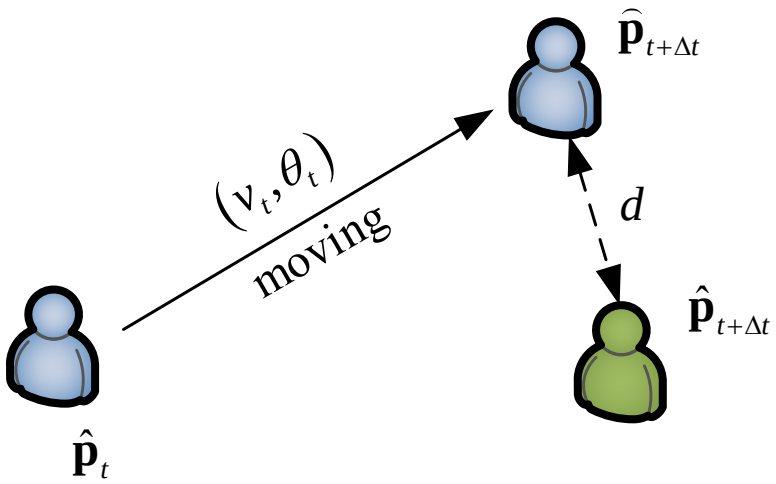


Figure 78 Illustration of MSI assisted model monitoring

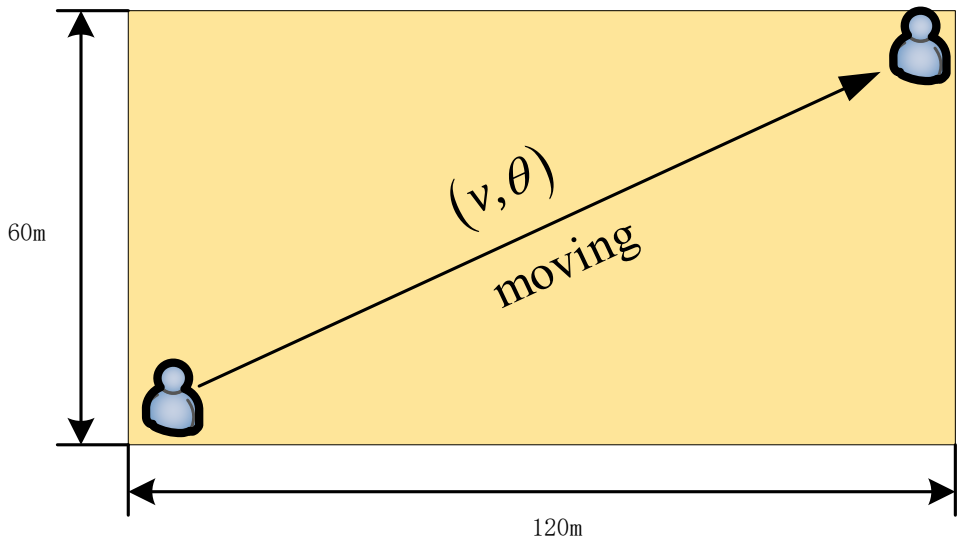


Figure 79 Illustration of trajectory design

When it comes to simulations, as shown in Figure 79, we design a simple trajectory from the lower left to the upper right corner of the indoor factory. The terminal moves with a fixed speed $v = 3km/s$, and records its MSI every $\Delta t = 0.5s$. We compare the distributions of distance d for the two cases. As a comparison, we firstly evaluate the distribution of distance d when training dataset for model training and test dataset for model monitoring come from different drops. The case in which training dataset for model training and test dataset for model monitoring come from the same drop is adopted as a reference case. As shown in Figure 80, the distance d for the first case is very large, and the distribution of distance d for the first case significantly deviates from that of the reference case.

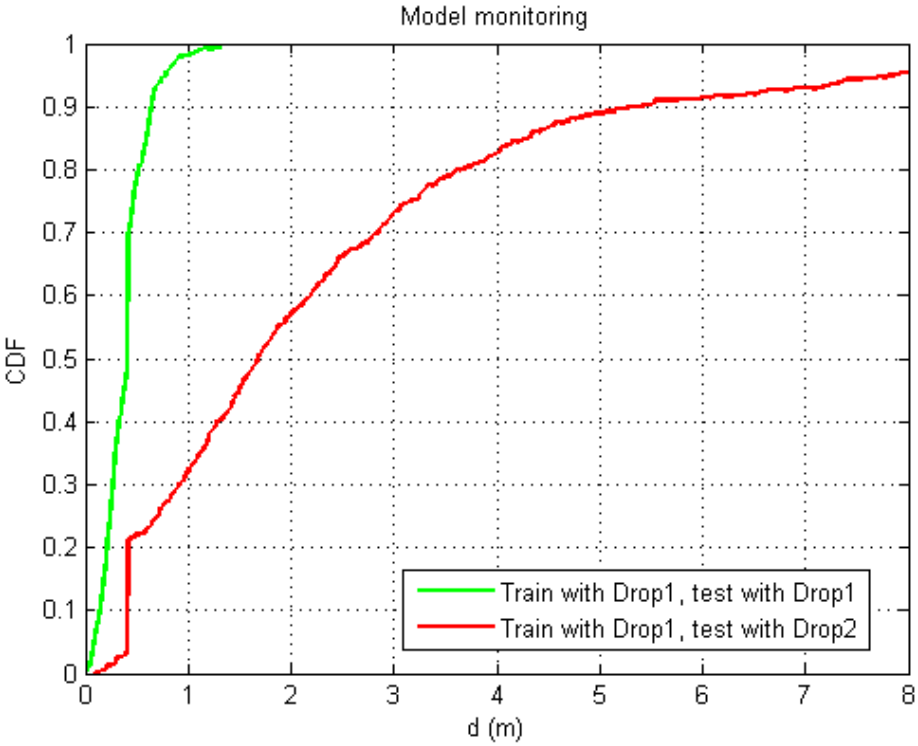


Figure 80 Illustration of simulation result

Table 64 Evaluation of distance metrics between two distributions

Distribution for reference	Distribution for test	Maximal vertical distance	KL divergence
Distance d for case1 (train with drop1, test with drop1)	Distance d for case2 (train with drop1, test with drop2)	0.68	2.46

Observation 43: Motion sensors can be used to assist model monitoring.

Proposal 27: Further study how to use motion sensors' information to assist model monitoring.

7.2.3. Self-monitoring for AI/ML assisted positioning

At the RAN1 #112 meeting, it was agreed that:

- Agreement

For AI/ML **assisted** approach, study the performance of label-free model monitoring methods, which do not require ground truth label (or its approximation) for model monitoring.

In addition to the external information aforementioned, including PRU's location and sensor measurement information, additional redundant information from the positioning process can also be used to assist model monitoring, which we call *self-monitoring*. For example, the location of TRP is a kind of redundant information for AI/ML assisted positioning methods, which can be used to assist model monitoring.

Here, we propose a self-monitoring based model monitoring scheme for AI/ML assisted positioning. As illustrated in Figure 81, the details are presented as follows:

- N CIR vectors associated with N TRPs are measured and input into the AI/ML model for TOA estimation. Then, a set TOA1 containing N TOAs is obtained.
- Utilizing non-AI timing based positioning algorithm, the location of the target UE can be estimated based on the set TOA1 and related TRPs' location.
- Based on the estimated UE's location and TRPs' location, a set TOA2 containing N TOAs is calculated reversely.
- Calculate the Euclidean distance $d = \|TOA_1 - TOA_2\|_2$ between the set TOA1 and the set TOA2. When the distribution of d exceeds a predefined threshold, it means the current AI/ML model is invalid. Otherwise, the current AI/ML model still works.

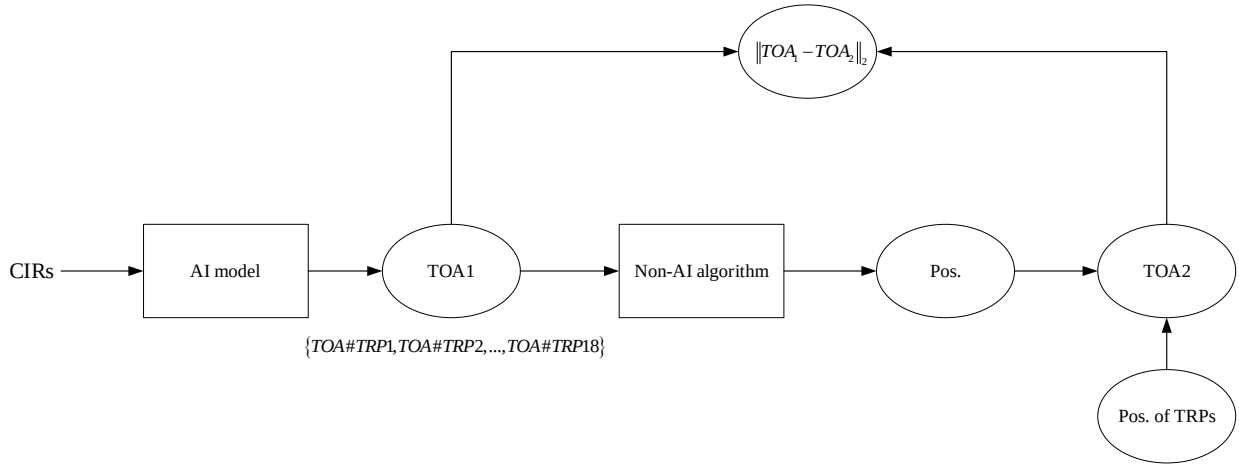


Figure 81 Illustration of self-monitoring for AI/ML assisted positioning

When it comes to simulations, the case where the training and test datasets come from the same drop is adopted as a reference case, and we further evaluate the case where the training and test datasets come from different drops. As shown in Figure 82, it is observed that the distance d of the reference case (blue line) is greatly smaller than that of the evaluation case (red line). Meanwhile, the distribution of distance d for the evaluation case significantly deviates from that of the reference case.

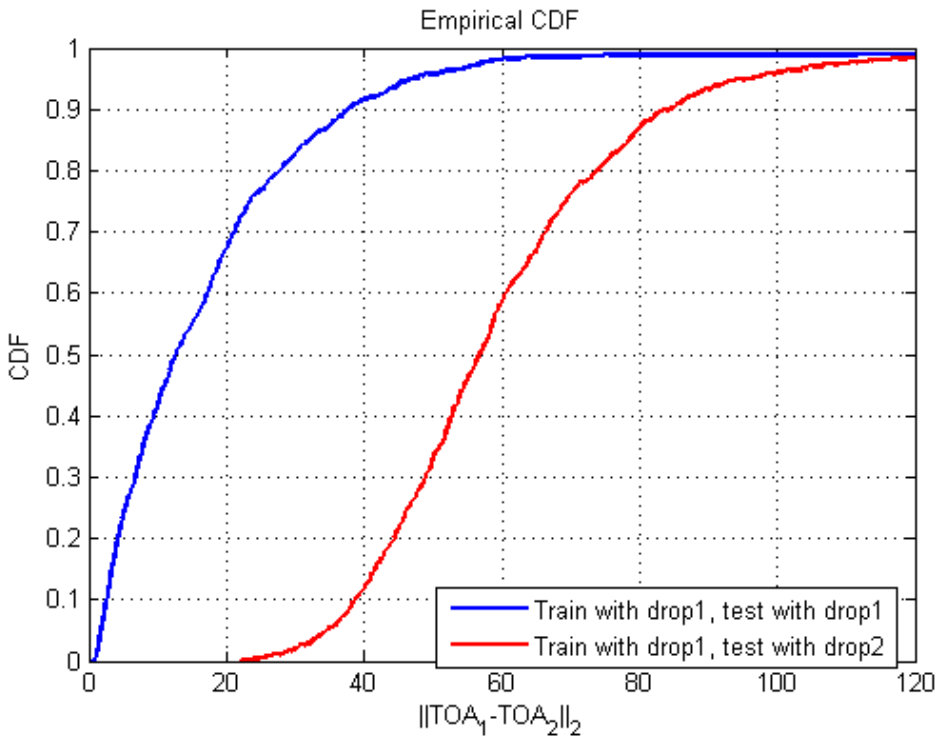


Figure 82 Illustration of simulation result

Table 65 Evaluation of distance metrics between two distributions

Distribution for reference	Distribution for test	Maximal vertical distance	KL divergence
Distance d for case1 (train with drop1, test with drop1)	Distance d for case2 (train with drop1, test with drop2)	0.82	36.94

Observation 44: The proposed self-monitoring method can achieve model monitoring for AI/ML assisted positioning.

Proposal 28: Further study the model monitoring schemes for AI/ML assisted positioning.

7.2.4. Ranging model assisted model monitoring

Section 7.2.2 presents a displacement distance-based model monitoring method, where the displacement distance is measured by motion sensor equipped at UE side. Moreover, such distance measurement can also come from an AI/ML model. Specifically, when training an AI/ML for positioning, another ranging model is also generated based on the same dataset. Given two CIRs as model input, the ranging model can output their distance directly which is equal to their distance in physical location. Essentially, the ranging model builds a mapping from CIR space to another latent space, and the distance between two UEs in latent space is equal to that in physical space.

$$d_1 = \left\| g\left(h_1\right) - g\left(h_2\right) \right\|$$
$$d_2 = \left\| f\left(h_1\right) - f\left(h_2\right) \right\|$$
$$d_1 = d_2$$

Where h_1 and h_2 denote CIRs of UE1 and UE2, $g\left(\bullet\right)$ is the neural network function for positioning, and $f\left(\bullet\right)$ is the neural network function for ranging. In theory, when the positioning model works well, the distance between two UEs in latent space should be equal to that in physical space, i.e., $d_1 = d_2$. Otherwise, the distance between two UEs in latent space will severely deviate from that in physical space, i.e., $\left|d_1 - d_2\right| > \varepsilon$. Figure 83 presents the training process of ranging model, which is implemented based on Siamese network.

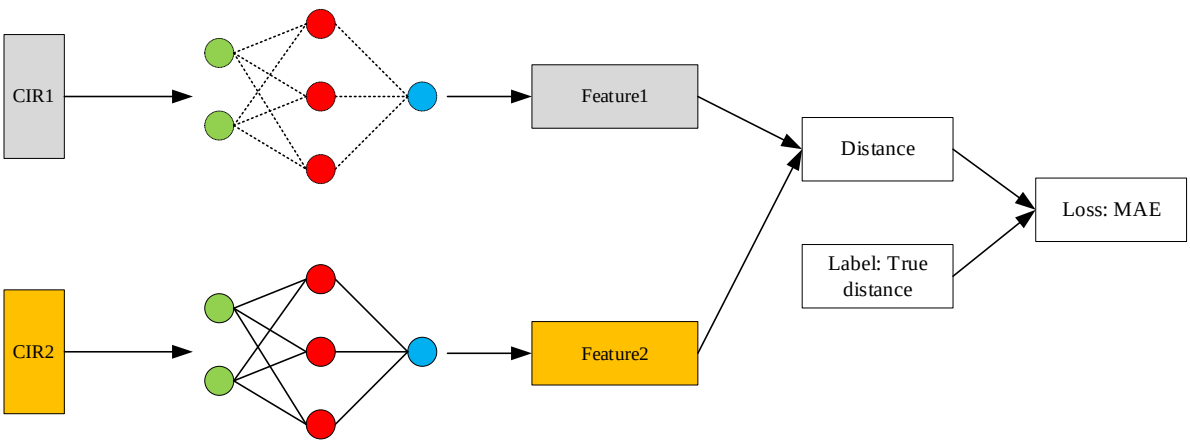


Figure 83 Illustration of the training process of ranging model

Figure 84 presents how to calculate a metric or distance for model monitoring with the assistance of ranging model, which is illustrated as follows:

- Obtain two instances (CIR). These two instances may come from two different UEs or two different slots of the same UE.
- Input two instances into the positioning model and estimate their distance d_1 in physical space.
- Input two instances into the ranging model and estimate their distance d_2 in latent space.

- Calculate the metric for model monitoring according to d_1 and d_2, e.g., $d = |d_1 - d_2|$.

When there are many pairs of instances, we can make the decision on model monitoring according to the distribution of the metric d. As shown in Figure 85, it is observed that when the tested drop of the positioning model is consistent with training dataset, the metric d is less than 0.5m@90%. On the contrary, the metric d is greatly larger than 10m@90%. Thus, ranging model assisted model monitoring is valid for direct AI/ML positioning without need of ground truth label.

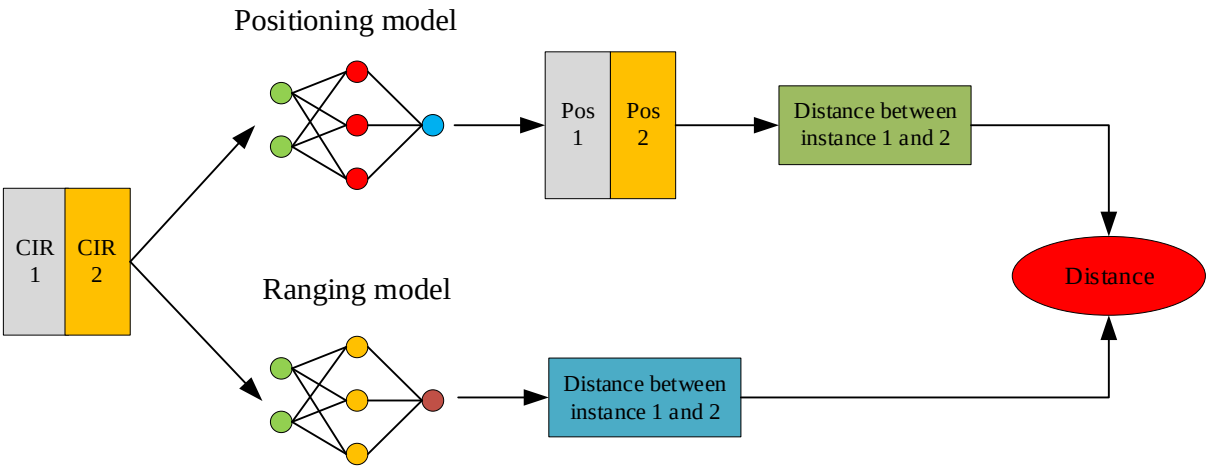


Figure 84 Illustration of model monitoring with the assistance of ranging model

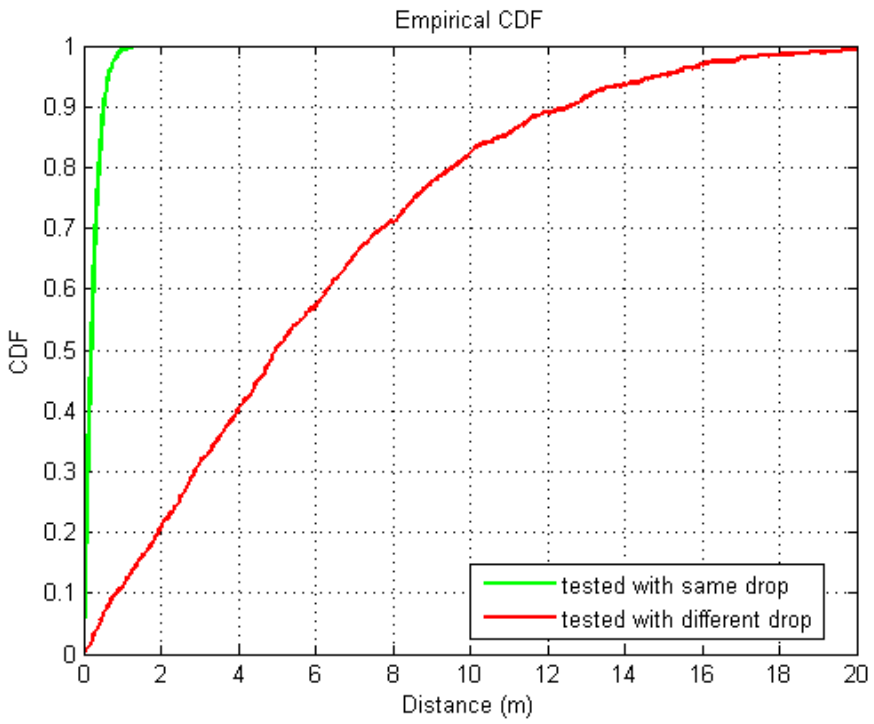


Figure 85 Illustration of simulation result

Table 66 Evaluation of distance metrics between two distributions

Distribution for reference	Distribution for test	Maximal vertical distance	KL divergence
Distance d for case1 (train with drop1, test with drop1)	Distance d for case2 (train with drop1, test with drop2)	0.88	3.79

Observation 45: Ranging model assisted model monitoring is valid for direct AI/ML positioning without need of ground truth label.

7.3. Assessment of model monitoring

In this section, we have presented several model monitoring methods from the perspective of the sources of assistance information, including model input and model output, and more model monitoring schemes are emerging. How to assess these model monitoring schemes is also an open issue at present. In our view, the first step is to align the simulation assumptions for model monitoring, and testing AI/ML model with another different drop could be a starting point. As for the assessment of model monitoring, the following two aspects could be considered as a starting point

- Feasibility: The feasibility consists of the accessibility of reference information. The reference information, which is used to compare with model input or output related information, may come from source training dataset of monitored AI/ML model or measurement. For the model output based model monitoring methods, the reference information comes from the measurement. An example of model output based model monitoring is illustrated as Figure 86. For example, for the ground truth label based model monitoring method, the reference information is ground truth label, coming from PRU or UE measurement. For the motion sensor assisted model monitoring method, the reference information is UE's motion information, coming from UE' sensor measurement. For the model input based model monitoring methods, the reference information comes from the source training dataset. An example of model output based model monitoring is illustrated as Figure 87. For example, for the shift detection of dominant feature distribution, the reference information is the feature distribution, extracted from the source training dataset. For the AI/ML based output-of-distribution detection, the reference information is the AI/ML model used for out-of-distribution detection, and the AI/ML model is generated via the source training dataset.
- Performance: It is agreed that the decision of model monitoring should be made based on statistic information, rather than only relying on single or several instances. The distance between the distribution of model input or output related information and the distribution of the reference information could be adopted as the performance metric of model monitoring. In general, under the same simulation setup, the larger the distance between these two distributions, the easier it is for the model monitoring method to identify the failure of AI/ML model.

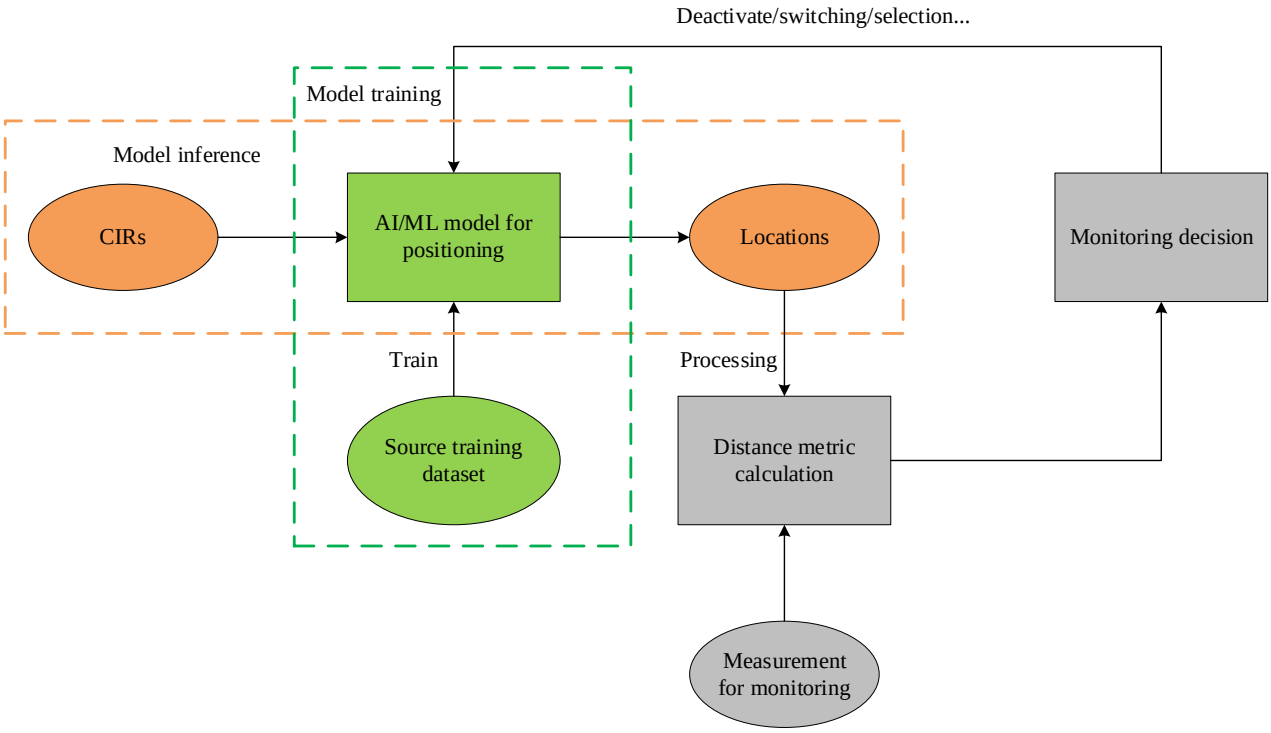


Figure 86 An example of model output based model monitoring

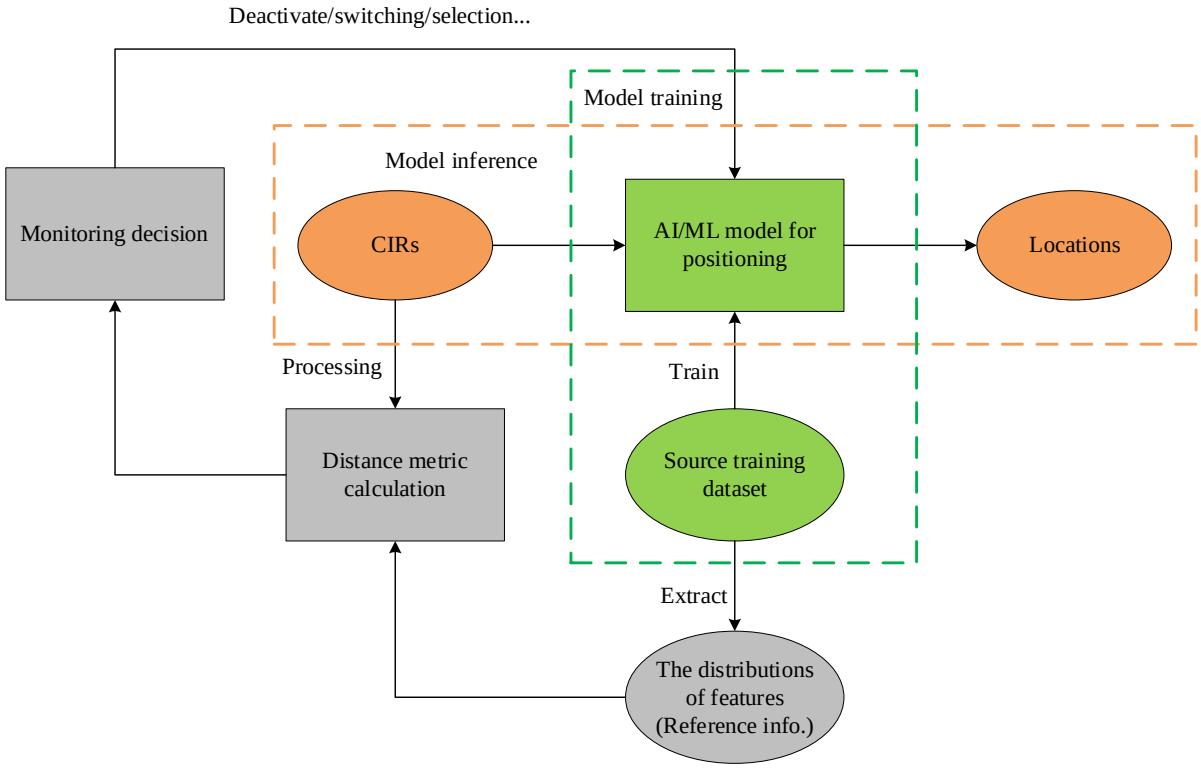


Figure 87 An example of model input based model monitoring

Proposal 29: Testing AI/ML model with another different drop could be a starting point for the evaluation of model monitoring methods.

Proposal 30: For the assessment of model monitoring, at least feasibility and performance should be considered as a starting point.

8. Performance evaluation for reduced CIR/PDP size

Compared with legacy positioning methods, AI/ML based positioning adopting CIR/PDP as model input may consume huge air-interface resource to report CIR/PDP measurement from UE side to NW side, such as for Case 2b. According to the existing experience of CSI reporting specified, reporting CIR/PDP after compression will significantly reduce the bit overhead at the cost of possible performance degradation. Hence, we study some simple CIR/PDP compression methods including truncating CIR/PDP in time domain and reducing the number of TRPs, and further evaluate their positioning performance and required bit overhead as compared with legacy RSRPP reporting.

8.1. Truncating CIR/PDP in time domain

In general, time-domain CIR can be obtained by an IFFT of frequency-domain CIR which is estimated by measuring reference signal, such as CSI-RS and PRS. For a typical transmission configuration of bandwidth 100MHz and subcarrier space 30kHz, the size of frequency-domain sampling points for an OFDM symbol is 4096 so as to reach a sampling rate > 100MHz. As a result, the size of IFFT window transforming frequency-domain CIR to time domain CIR is also 4096. Real-time reporting of a 4096-point time-domain CIR will impose a substantial burden on the air-interface resource. Consequently, truncating CIR in time domain is a feasible solution to mitigate this issue. As shown in Figure 88, only first 30 samples of time-domain CIR are retained and the remaining part is discarded.

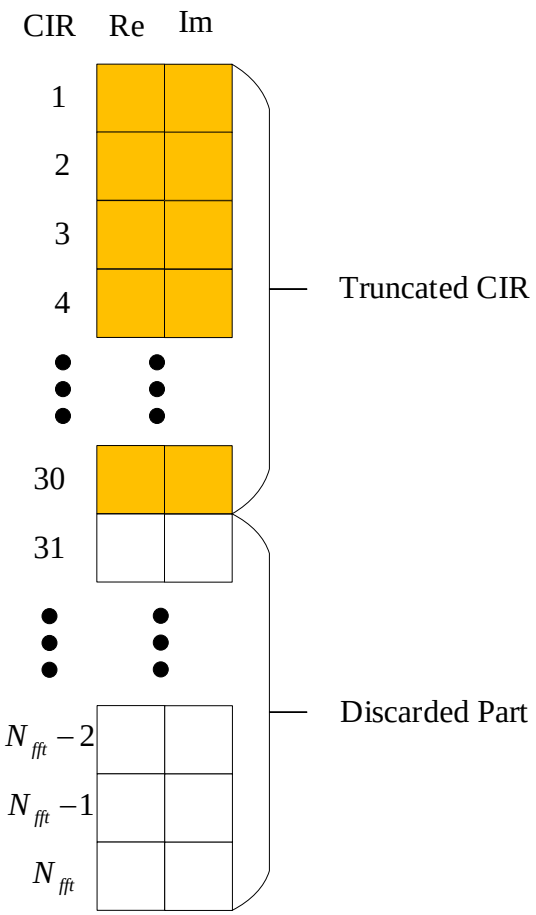


Figure 88 Illustration of truncated time-domain CIR

8.1.1. Direct AI/ML positioning

How CIR/PDP truncation affects the positioning performance for direct AI/ML positioning? From Table 67, it is observed that the performance degradation is negligible when only the first 30 samples of time-domain CIR are retained. However, the positioning performance is severely degraded when the 30th to 100th samples are retained, which indicates that the position of truncation has a significant impact on positioning performance and the first 30 samples contain more information related to UE's location for direct AI/ML positioning.

Table 67 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Truncated CIR**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	1-30	DH	DH	25k	1k	1.65M	22.30M	1.04
CIR	Pos.	1-100	DH	DH	25k	1k	1.65M	22.30M	1.04
CIR	Pos.	30-100	DH	DH	25k	1k	1.65M	22.30M	2.99

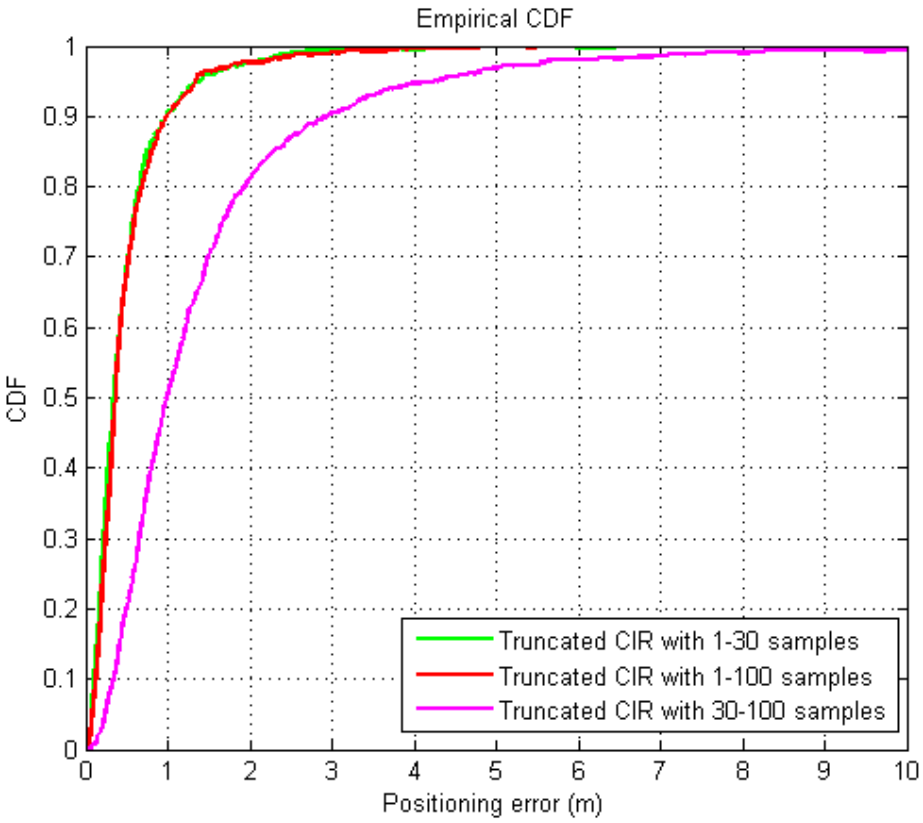


Figure 89 CDF of positioning accuracy with different number of truncated CIR

Observation 46: There is no obvious positioning accuracy degradation for direct AI/ML positioning when only the first 30 samples of time-domain CIR are truncated.

Observation 47: The position of truncation of CIR has a significant impact on positioning performance for direct AI/ML positioning.

Proposal 31: Further study the impact of CIR truncation on positioning performance and overhead reduction for direct AI/ML positioning, at least for the following aspects:

- **The minimal length of truncated CIR to reach a target positioning accuracy.**
- **The position of truncated CIR, such as the front part or the middle part.**

8.1.2. AI/ML assisted positioning

How CIR/PDP truncation affects the positioning performance for AI/ML assisted positioning? From Table 68, different from direct AI/ML positioning, CIR truncation causes obvious performance degradation for AI/ML assisted positioning. For example, truncating the first 30 samples of time-domain CIR will not cause obvious performance degradation for direct AI/ML positioning with performance dropping from 0.99m@90% to 1.04m@90%, while severely impairs the positioning performance for AI/ML assisted positioning with accuracy dropping from 0.73m@90% to 3.72m@90%. This reason behind is that AI/ML based TOA estimation also relies on the fingerprint feature of CIR for these heavy-NLOS scenarios, while CIR truncation severely reduces the information contained in the fingerprint feature when only one-TRP CIR is adopted as model input. Therefore, for AI/ML assisted positioning, longer CIR is needed to achieve high-accuracy positioning comparable to direct AI/ML positioning.

Table 68 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Truncated CIR**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML

CIR	TOA	1-256	DH	DH	25k	1k	4.26M*18	8.50M*18	0.73
CIR	TOA	1-30	DH	DH	25k	1k	4.26M*18	8.50M*18	3.72
CIR	TOA	1-100	DH	DH	25k	1k	4.26M*18	8.50M*18	1.04
CIR	TOA	30-100	DH	DH	25k	1k	4.26M*18	8.50M*18	4.17

Table 69 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Truncated CIR**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m) CDF = 90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	1-256	DH	DH	25k	1k	4.26M*18	8.50M*18	0.62
CIR	TOA	1-30	DH	DH	25k	1k	4.26M*18	8.50M*18	3.25
CIR	TOA	1-100	DH	DH	25k	1k	4.26M*18	8.50M*18	0.98
CIR	TOA	30-100	DH	DH	25k	1k	4.26M*18	8.50M*18	3.40

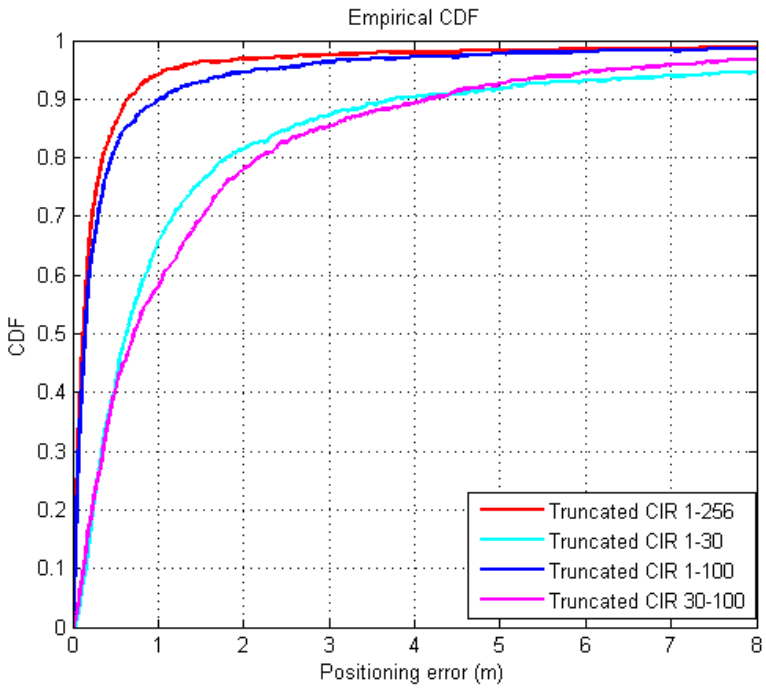


Figure 90 CDF of positioning accuracy with different number of truncated CIR

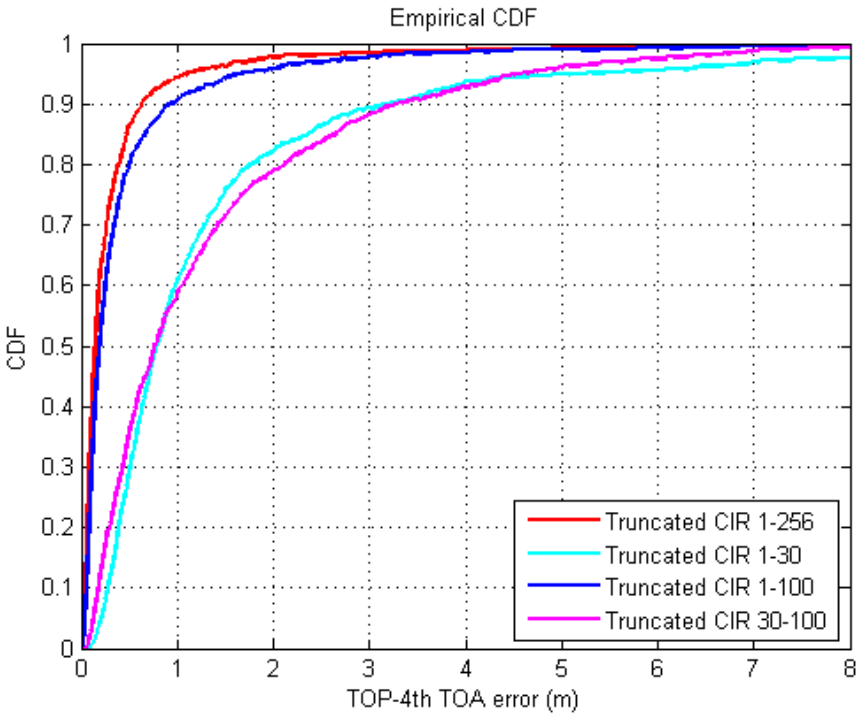


Figure 91 CDF of ToA accuracy with different number of truncated CIR

Observation 48: For AI/ML assisted positioning, longer CIR is needed to achieve high-accuracy positioning comparable to direct AI/ML positioning.

8.2. Reducing the number of TRPs

The agreed simulation assumption is that 18 TRPs are deployed at an indoor factory for AI/ML based positioning. Actually, when considering the actual model deployment, it is difficult to guarantee that 18 TRPs are always available for positioning since the number of TRPs is different for different areas. Moreover, in addition to reducing the length of time-domain CIR, reducing the number of TRPs used for positioning can also reduce the overhead of CIR measurement and reporting at the cost of possible performance degradation, as illustrated in Figure 92. In this subsection, we further evaluate the impact of the number of TRPs on positioning performance, expecting to obtain some observations on the trade-off between the number of TRPs and positioning performance for both direct AI/ML positioning and AI/ML assisted positioning.

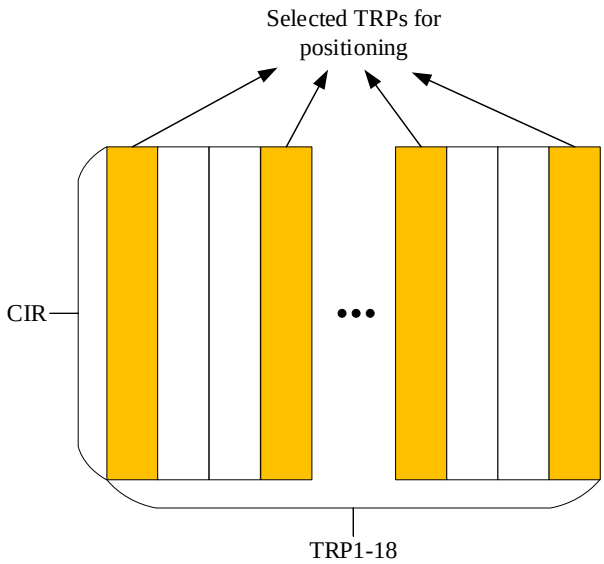


Figure 92 Illustration of reducing the number of TRPs for positioning

8.2.1. Direct AI/ML positioning

As shown in Table 70, it is indicated that reducing the number of TRPs used for positioning can slightly degrade the positioning accuracy for direct AI/ML positioning. For examples, when reducing the number of TRPs from 18 to 4, the positioning accuracy degrades from 0.99m@90% to 1.60m@90%. However, the positioning accuracy will be degraded severely when further reducing the number of TRPs from 4 to 2. Therefore, the minimal number of TRPs used for positioning should be identified to reach the target positioning accuracy for direct AI/ML positioning.

Table 70 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**Number of TRP**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	2	DH	DH	25k	1k	1.65M	22.30M	4.71
CIR	Pos.	4	DH	DH	25k	1k	1.65M	22.30M	1.60
CIR	Pos.	9	DH	DH	25k	1k	1.65M	22.30M	1.26
CIR	Pos.	18	DH	DH	25k	1k	1.65M	22.30M	0.99

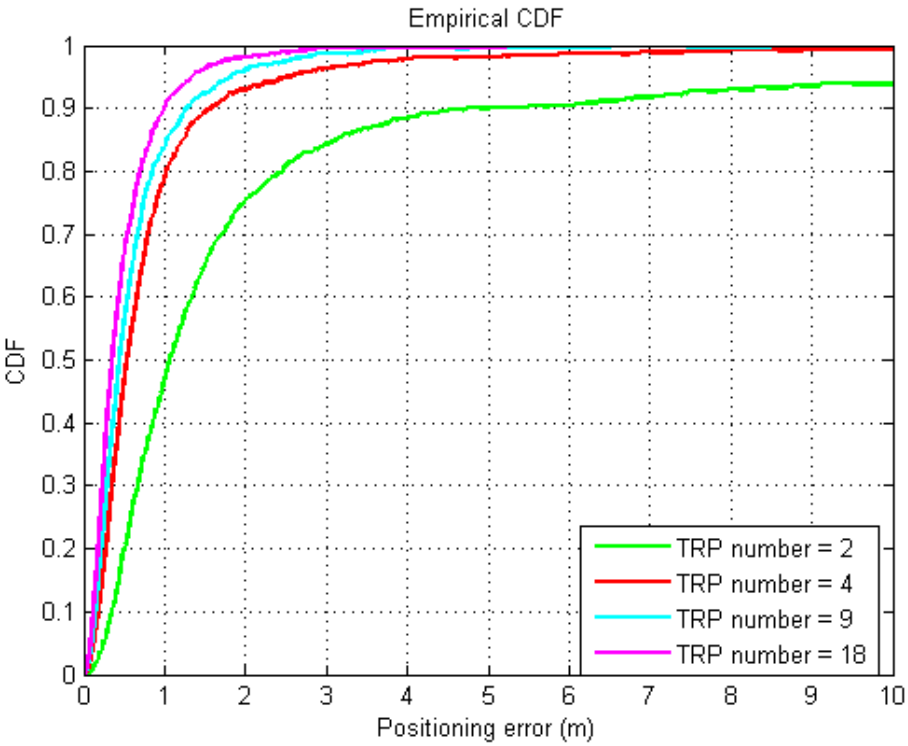


Figure 93 Illustration of simulation result

Observation 49: Reducing the number of TRPs used for positioning can slightly degrade the positioning accuracy but still acceptable.

Proposal 32: Further study the impact of the number of TRPs on positioning accuracy and identify the minimal number of TRPs used for positioning to ensure the target positioning accuracy for direct AI/ML positioning.

8.2.2. AI/ML assisted positioning

Comparing Table 70 with Table 71, we observe that AI//ML assisted positioning is more sensitive to the number of TRPs used for positioning than direct AI/ML positioning. For example, when there are only four TRPs available for positioning, the positioning accuracy of direct AI/ML positioning is 1.6m@90%, greatly higher than that of

AI/ML assisted positioning (5.76m@90%). However, with the increase of TRP number, the positioning accuracy of AI/ML assisted positioning is increasingly close to, and even surpassing that of direct AI/ML positioning. Consequently, the minimal number of TRPs used for positioning should be identified to reach the target positioning accuracy for AI/ML assisted positioning.

Table 71 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**TRP number**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	4	DH	DH	25k	1k	1.65M	22.30M	5.76
CIR	TOA	9	DH	DH	25k	1k	1.65M	22.30M	1.50
CIR	TOA	18	DH	DH	25k	1k	1.65M	22.30M	0.73

Table 72 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**TRP number**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**TOP-4th TOA accuracy (m) CDF = 90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	TOA	4	DH	DH	25k	1k	1.65M	22.30M	9.85
CIR	TOA	9	DH	DH	25k	1k	1.65M	22.30M	1.97
CIR	TOA	18	DH	DH	25k	1k	1.65M	22.30M	0.62

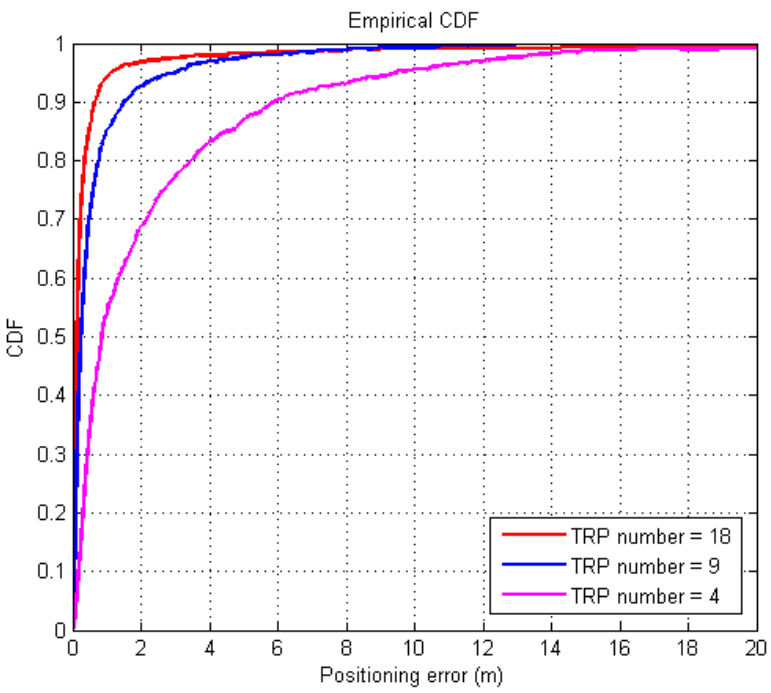


Figure 94 CDF of positioning accuracy

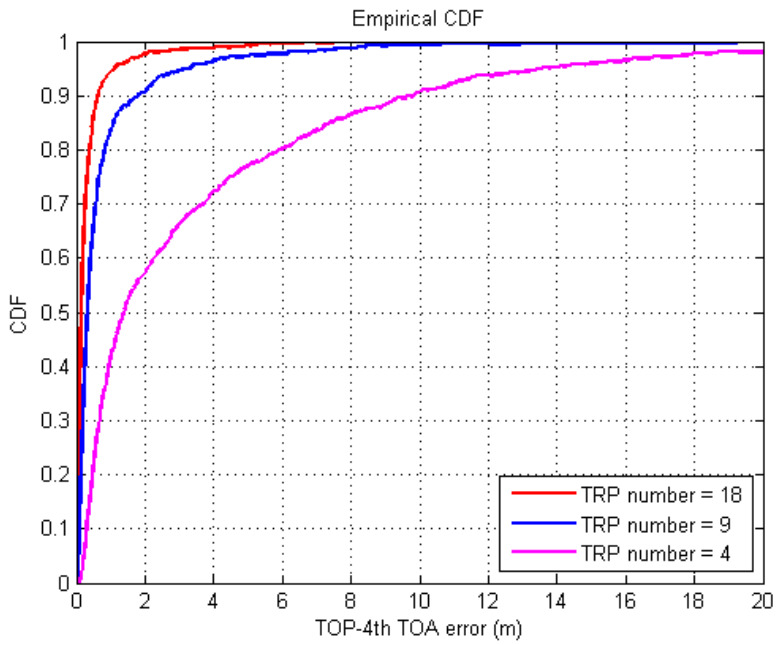


Figure 95 CDF of accuracy of TOA estimation

Observation 50: AI//ML assisted positioning is more sensitive to the number of TRPs used for positioning than direct AI/ML positioning

Proposal 33: The minimal number of TRPs used for positioning should be identified to reach the target positioning accuracy for AI/ML assisted positioning.

Observation 51: Direct AI/ML positioning has greater potential for compressing the dimensions of the CIR/PDP compared to AI/ML assisted positioning.

Proposal 34: For direct AI/ML positioning, support to report CIR/PDP from UE side to NW side.

8.3. Evaluation of overhead and performance for CIR/PDP and RSRPP reporting

Apart from reducing the length of CIR/PDP and the number of TRPs, there are also other methods to reduce the bit overhead of CIR/PDP report, such as utilizing the structure information of CIR/PDP, quantization and so on. For PDP/CIR reporting without compression, the required number of bits is calculated as follows:

$$N_{bits} = N_{TRP} \times N_{tx_port} \times N_{rx_port} \times N_{cir} \times N_{quantization} \times 2$$

where $N_{TRP}, N_{tx_port}, N_{rx_port}, N_{cir}, N_{quantization}$ denote the number of TRPs, transmitting ports, receiving ports, CIR points and quantization bits, respectively. The term '2' denotes the real and imaginary part of a complex value. Here, we assume the single-port positioning is adopted, i.e., $N_{tx_port} = N_{rx_port} = 1$. From this formula, it is also confirmed that reducing the number of TRPs and the length of time-domain CIR can linearly reduce the bit overhead of CIR reporting.

Moreover, we find that only a small amount of multipath information can be captured into time-domain CIR, and such sparsity feature can also be used to further compress the overhead of PDP/CIR reporting. In particular, many CIRs after truncation are zero vectors, especially for these users that are far away from TRPs. Therefore, we propose the following CIR compression method:

- Part 1: 1 bit is used to indicate that whether this CIR is a zero vector.
- Part2: If not a zero vector, report the positions of the non-zero elements within CIR.
- Part3: If not a zero vector, report the values of the non-zero elements within CIR.

It is noted that the values of the non-zero elements should be quantified before reporting to NW side. This method can be also used for PDP compression. Specifically, for CIR and PDP reporting, the bit overhead after compression is calculated as follows:

$$N_{bits} = N_{part1} + N_{part2} + N_{part3}$$

Where

$$N_{part1} = 1$$

$$N_{part2} = \lceil \log_2 N_{non_zero_cir} \rceil$$

- For CIR, each non-zero element is a complex number, and both real part and imaginary part should be included:

$$N_{part3} = N_{TRP} \times N_{tx_port} \times N_{rx_port} \times N_{non_zero_cir} \times N_{quantization} \times 2$$

- For PDP, each non-zero element is a real number, and only one part should be included:

$$N_{part3} = N_{TRP} \times N_{tx_port} \times N_{rx_port} \times N_{non_zero_cir} \times N_{quantization}$$

Where $N_{non_zero_cir}$ is the number of non-zero elements within a CIR.

Existing specifications have supported RSRPP reporting of up to 8 paths for each TRP. $N_{RSRPP} = 8$, and the overhead is calculated as follows:

- RSRPP reporting without path delay:

$$N_{bits} = N_{TRP} \times N_{quantization} \times N_{RSRPP}$$

- RSRPP reporting with path delay, and path delay is quantized by 8 bits (ranging from 1~256 time-domain samples in CIR):

$$N_{bits} = N_{TRP} \times N_{quantization} \times N_{RSRPP} + N_{TRP} \times N_{RSRPP} \times 8$$

Taking legacy RSRPP based positioning as baseline, we evaluate the overhead (in terms of number of bits) of different measurement reporting for AI/ML based positioning. Assume that $N_{TRP} = 4, 9, 18$, $N_{tx_port} = N_{rx_port} = 1$, $N_{quantization} = 8$, and the first 32 samples of CIR/PDP are truncated. As shown in Figure 96, the overhead of PDP reporting is slightly larger than that of RSRPP reporting without delay (only report 8 RSRPPs without their position in time-domain); and the overhead of RSRPP reporting with delay is greatly larger than that of PDP reporting and slightly larger than that of CIR reporting. With the increase of TRP number, the overhead linearly increases. Therefore, it is concluded that the overhead of CIR/PDP reporting is acceptable as compared with the legacy RSRPP reporting.

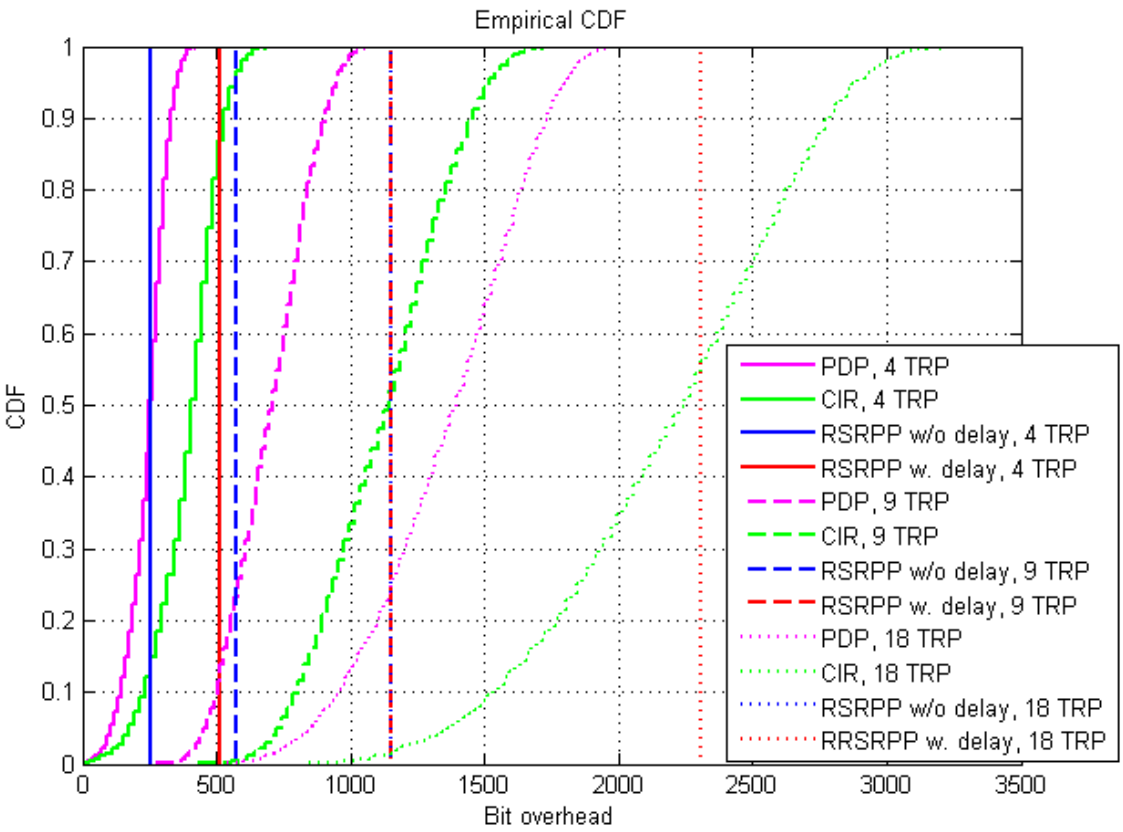


Figure 96 Evaluation of overhead of different measurement reports

Moreover, we also compare the positioning performance of RSRPP based positioning and CIR based positioning when RSRPP and PDP are adopted as model input, respectively. As shown in Table 73, it is shown that adopting CIR as model input reaps greatly better positioning performance as compared with adopting 8 RSRPPs as model input when there exists 18 TRPs available, and the positioning performance of adopting RSRPPs of the first 8 paths as model input is slightly better than that of adopting 8 RSRPPs with the maximal power. When reducing the number of TRPs, the overhead is significantly reduced for CIR reporting and even less than legacy RSRPP reporting, but the positioning performance of adopting CIR as model input is still greatly better than or at least comparable with that of adopting RSRPP as model input. For example, the bit overhead of 9 TRPs' CIRs reporting is similar to that of 18 TRPs' RSPRPs reporting without delay, but the positioning performance of the former 1.34m@90% is obviously better than that of the latter 2.14m@90%. The bit overhead of 4 TRPs' CIRs reporting is significantly less than that of 18 TRPs' RSPRPs reporting without delay, but their positioning performance is still comparable. Therefore, it is concluded that adopting CIR as model input can reap greatly better positioning performance as compared with adopting RSRPP as model input with acceptable bit overhead.

Table 73 Evaluation results for AI/ML model deployed on UE or Network side, UE distribution area = [120x60 m]

Model input	**Model output**	**TRP number**	**Truncated CIR**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
				Train	Test	Train	test	Model complexity	Computation complexity	AI/ML
CIR	Pos.	18	1-32	DH	DH	25k	1k	1.65M	22.30M	0.99
CIR	Pos.	9	1-32	DH	DH	25k	1k	1.65M	22.30M	1.34
CIR	Pos.	4	1-32	DH	DH	25k	1k	1.65M	22.30M	2.53

RSRPP of first 8 paths	Pos.	18	/	DH	DH	25k	1k	1.65M	22.30M	2.14
RSRPP of top 8 paths	Pos.	18	/	DH	DH	25k	1k	1.65M	22.30M	2.39

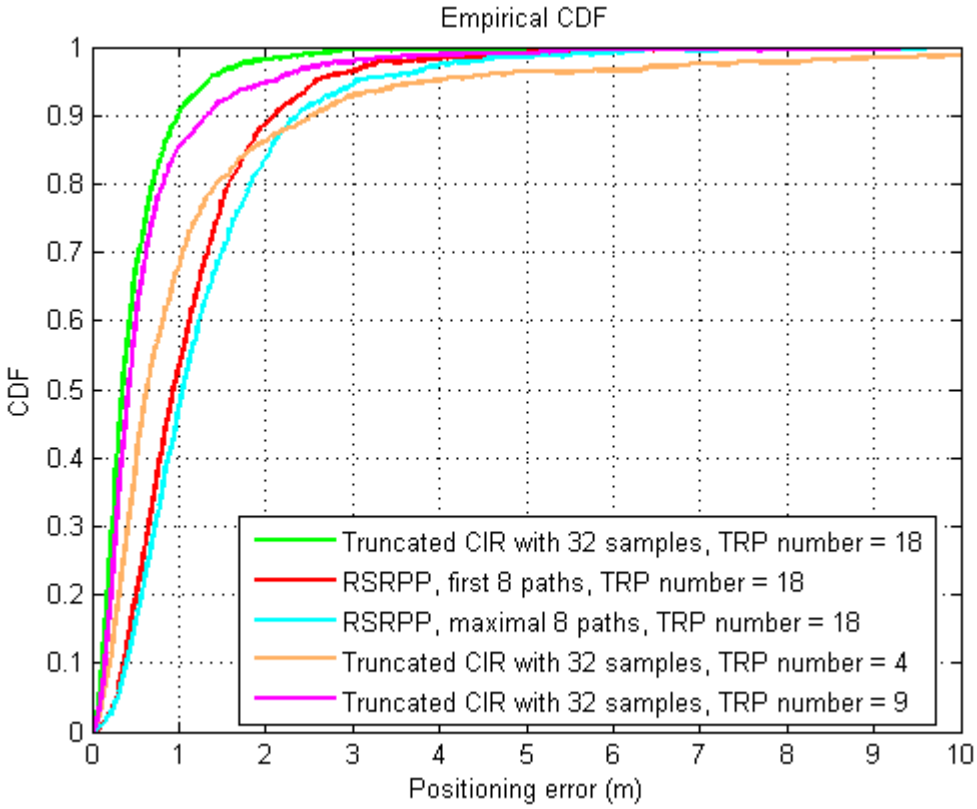


Figure 97 Illustration of simulation result

Observation 52: The bit overhead of CIR/PDP reporting is slightly larger than legacy RSRPP reporting but still acceptable.

Observation 53: Adopting CIR as model input can reap greatly better positioning performance as compared with adopting RSRPP as model input.

Proposal 35: Further study CIR/PDP compression to support CIR/PDP reporting.

9. Cost evaluation

In the previous sections, we mainly evaluate the positioning accuracy performance and generalization capability for AI/ML based positioning, and observe that AI technology has great potential to improve positioning accuracy. On the other hand, power consumption, computational complexity, parameter quantity, training data requirements and hardware requirements (including for given processing delays) associated with enabling respective AI/ML scheme are essential for practical deployment of AI based positioning. Here, we mainly focus on computational complexity, parameter quantity, training data requirements in this section.

9.1. Model assumption

In light of "All models are wrong, but some models are useful, George Box", we think the selection of AI/ML model may be strongly related to specific tasks, and a suitable model can facilitate better evaluation of performance gain for AI/ML based positioning. We adopt two different AI models, vision transformer and full-connection neural network (FNN), to evaluate the positioning performance of multi-TRPs and single-TRP based positioning methods, respectively.

9.2. Cost related KPIs

Agreement

For evaluation of AI/ML based positioning, the model complexity is reported via the metric of "number of model parameters".

The cost related KPIs of these used models are listed in Appendix B.

Table 74 Cost evaluation of AI models

AI models	FNN1	FNN2	Vision Transformer
Computational complexity	4.26M FLOPs*18	11.92M*1	22.30M FLOPs*1
Number of Parameters	8.50M*18	23.79M*1	1.65M*1

The computational complexity and parameter quantity can be further reduced by model optimization, and some other models may get better performance with lower cost.

10. Conclusions

In this contribution, we discuss AI/ML based positioning accuracy enhancement with the following observations.

Observation 1: AI/ML based positioning can significantly improve the positioning accuracy compared to existing RAT-dependent positioning methods in heavy NLOS scenarios.

Observation 2: Different inputs of AI model will affect the positioning performance for AI/ML based positioning. Time domain channel CIR as the input of AI model obtains the best positioning accuracy.

Observation 3: For Construction 1 (Single-TRP, N models for N TRPs), the AI/ML based TOA estimation positioning method achieves remarkable performance gain compared to direct AI/ML positioning method.

Observation 4: For Construction 2 (Single-TRP, same model for N TRPs), it is beneficial to include TRP's information into model input so as to improve the positioning accuracy.

Observation 5: ToA as an intermediate report is better than RSTD for AI/ML assisted positioning.

Observation 6: AI/ML based LOS/NLOS identification for positioning has the following advantages:

- **More accurate LOS/NLOS identification along with a confidence metric**
- **Better compatibility with existing positioning protocol framework.**
- **Great generalization capability.**

and disadvantages:

- **Positioning performance could suffer from severe degradation in heavy-NLOS scenarios.**
- **Obtain LOS/NLOS labels is a challenging task for data collection.**

Observation 7: Positioning performance of direct AI/ML positioning degrades when the model trained with dataset of one drop is tested with dataset of other drops.

Observation 8: Positioning performance of direct AI/ML positioning degrades when the training and testing datasets are of different clutter parameters in an InF-DH scenario.

Observation 9: Training AI/ML model with a mixed dataset is an effective way to improve model generalization performance.

Observation 10: The positioning accuracy of direct AI/ML positioning trained with dataset from one InF scenario is seriously degraded when tested on dataset from a different InF scenario.

Observation 11: Positioning performance of AI/ML assisted positioning degrades when the model trained with dataset of one drop is tested with dataset of other drops.

Observation 12: Positioning performance of AI/ML assisted positioning is slightly degraded but still acceptable when the model trained with dataset of one clutter parameter is tested with dataset of another clutter parameter.

Observation 13: AI/ML assisted positioning enjoys better generalization performance than direct AI/ML positioning across clutter parameters.

Observation 14: Positioning performance of AI/ML assisted positioning is degraded when the model trained with dataset of DH is tested with datasets of SH and HH.

Observation 15: For those scenarios whose positioning does not rely on fingerprint features, AI/ML based TOA estimation has better generalization ability than direct AI/ML positioning.

Observation 16: AI/ML based TOA estimation has great advantages in positioning performance, deployment flexibility, compatibility with existing positioning protocol framework, and generalization capability.

Observation 17: The interference from TPRs can dramatically impair the positioning performance of AI/ML model.

Observation 18: The positioning accuracy of AI/ML based positioning significantly degrades with the increase of network synchronization error.

Observation 19: The positioning accuracy of AI/ML model is significantly improved from 10.18m@90% to 1.52m@90% by mix-training with samples of synchronization error.

Observation 20: The positioning accuracy gradually degrades with the increase of labeling error, but is still acceptable until standard deviation σ is 1 m (2.17m@90%). The maximum acceptable labeling errors (standard deviation) in the horizontal direction should be less than 1m to achieve 2m@90% positioning accuracy.

Observation 21: AI/ML based positioning is robust to label noise to some extent.

Observation 22: The positioning accuracy of AI/ML based positioning degrades with the increase of network synchronization error for direct AI/ML positioning.

Observation 23: The positioning accuracy of AI/ML model is significantly improved from 3.45m@90% to 2.04m@90% by mix-training with samples of UE timing error for direct AI/ML positioning.

Observation 24: CIR estimation error can dramatically degrade the positioning performance of AI/ML assisted positioning, while is more robust to small CIR estimation error compared to direct AI/ML positioning.

Observation 25: The positioning accuracy of AI/ML assisted positioning significantly degrades with the increase of network synchronization error, but it is still better than direct AI/ML positioning.

Observation 26: Labeling error can slightly impair the positioning performance for AI/ML assisted positioning, and AI/ML assisted positioning is more robust to labeling error compared with direct AI/ML positioning

Observation 27: Large UE timing error can dramatically deteriorate the positioning performance of AI/ML model for AI/ML assisted positioning, such as 50ns.

Observation 28: Fine-tuning the model with small amounts of samples from an unseen clutter parameter configuration can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such clutter parameter for direct AI/ML positioning.

Observation 29: Fine-tuning the model with small amounts of samples from an unseen drop can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to such new drop for direct AI/ML positioning.

Observation 30: Fine-tuning the model with small amounts of samples from an unseen scenario can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to such new scenario for direct AI/ML positioning.

Observation 31: Fine-tuning the model with small amounts of samples with an unseen synchronization error can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such synchronization error for direct AI/ML positioning.

Observation 32: Fine-tuning the model with small amounts of samples from an unseen clutter parameter can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to a new scenario with such clutter parameter for AI/ML assisted positioning.

Observation 33: Fine-tuning the model with small amounts of samples from an unseen drop can achieve significantly positioning accuracy improvement when the pre-trained model is transferred to such new drop for AI/ML assisted positioning.

Observation 34: The large-scale dataset is still required to fine-tune the pretrained model to the new environment for fingerprint based positioning, but has the advantages of reduced computational complexity compared with training an AI/ML model from scratch.

Observation 35: Fine-tuning the model with small amounts of samples from an unseen scenario can achieve huge positioning accuracy improvement when the pre-trained model is transferred to a new scenario for AI/ML assisted positioning

Observation 36: Fine-tuning the model with small amounts of samples with an unseen synchronization error can achieve obvious positioning accuracy improvement when the pre-trained model is transferred to a new scenario with a distinct synchronization error for AI/ML assisted positioning.

Observation 37: Model fine-tuning is suitable for the following tasks:

- **The source domain and the target domain are greatly similar, such as with different synchronization error.**
- **The target domain is easy to fit, such as TOA estimation of LOS path.**

Observation 38: Semi-supervised learning can achieve a more accurate position estimation as compared to supervised learning with less amount of labeled data.

Observation 39: Positioning with multi-port data can achieve a more accurate position estimation as compared to single-port positioning.

Observation 40: Adopting SINR as a dominant feature of CIR is valid for model monitoring

Observation 41: The proposed adversarial validation can achieve accurate model monitoring at the cost of acceptable hardware resource consumption for online training.

Observation 42: The proposed AI/ML based out-of-distribution detection can achieve accurate and flexible model monitoring without need of online training and large-scale data collection.

Observation 43: Motion state information can be used to assist model monitoring.

Observation 44: The proposed self-monitoring method can achieve model monitoring for AI/ML assisted positioning.

Observation 45: Ranging model assisted model monitoring is valid for direct AI/ML positioning without need of ground truth label.

Observation 46: There is no obvious positioning accuracy degradation for direct AI/ML positioning when only the first 30 samples of time-domain CIR are truncated.

Observation 47: The position of truncation of CIR has a significant impact on positioning performance for direct AI/ML positioning.

Observation 48: For AI/ML assisted positioning, longer CIR is needed to achieve high-accuracy positioning comparable to direct AI/ML positioning.

Observation 49: Reducing the number of TRPs used for positioning can slightly degrade the positioning accuracy but still acceptable.

Observation 50: AI//ML assisted positioning is more sensitive to the number of TRPs used for positioning than direct AI/ML positioning

Observation 51: Direct AI/ML positioning has greater potential for compressing the dimensions of the CIR/PDP compared to AI/ML assisted positioning.

Observation 52: The bit overhead of CIR/PDP reporting is slightly larger than legacy RSRPP reporting but still acceptable.

Observation 53: Adopting CIR as model input can reap greatly better positioning performance as compared with adopting RSRPP as model input.

We have the following proposals.

Proposal 1: Capture in the TR that time domain CIR as the model input for direct AI/ML positioning obtains the best performance compared to other model inputs.

Proposal 2: Support time domain CIR as the model input at least for direct AI/ML positioning.

Proposal 3: Support TOA as an intermediate measurement for reporting from UE side to LMF side or from gNB side to LMF side directly.

Proposal 4: Capture in the TR the benefits of AI/ML assisted positioning in terms of positioning accuracy and AI model generalization.

Proposal 5: Capture in the TR the benefits of training dataset with mixed/different configurations for AI/ML based positioning in terms of AI model generalization capability.

Proposal 6: Further study the impact and potential solution of CIR estimation error on AI/ML based positioning performance.

Proposal 7: Further study the impact and potential solution of network synchronization error on AI/ML based positioning performance.

Proposal 8: According to the requirement of positioning accuracy, the maximum acceptable labeling error should be identified firstly before data collection

Proposal 9: Further study the impact and potential solution of labeling error on AI/ML based positioning performance.

Proposal 10: Further study the impact and potential solution of UE timing error for direct AI/ML positioning.

Proposal 11: Study the solutions to mitigate the impact of CIR estimation error on positioning performance for AI/ML assisted positioning.

Proposal 12: Study the solutions to mitigate the impact of network synchronization error on positioning performance for AI/ML assisted positioning.

Proposal 13: Study the solutions to mitigate the impact of labeling error on positioning performance for AI/ML assisted positioning.

Proposal 14: Study the solutions to mitigate the impact of UE timing error on positioning performance for AI/ML assisted positioning.

Proposal 15: Further study and confirm the benefits of fine-tuning in terms of model generalization enhancement for direct AI/ML positioning.

Proposal 16: Further study and confirm the benefits of fine-tuning in terms of model generalization enhancement for AI/ML assisted positioning.

Proposal 17: Capture in the TR the benefits of fine-tuning for AI/ML assisted positioning in terms of positioning accuracy for AI model generalization capability.

Proposal 18: Both data efficiency and target performance could be considered as reference to determine the sample size required for model fine-tuning

Proposal 19: Model fine-tuning can flexibly adapt AI/ML model to various dynamic changes in environment, while mix-training is more suitable for these static changes.

Proposal 20: It is benefit to integrate model fine-tuning and mix-training so as to fully leverage their advantages.

Proposal 21: Capture in the TR the benefits of semi-supervised learning for AI/ML based positioning in terms of less data collection for training and more positioning accuracy.

Proposal 22: Capture in the TR the benefits of multi-port positioning for AI/ML based positioning in terms of positioning accuracy.

Proposal 23: When model input is CIR or PDP, identify these dominant features strongly related to positioning for model monitoring

Proposal 24: The metrics that can describe the difference between two distributions mathematically can be reused directly for model monitoring.

Proposal 25: Further study model input based model monitoring schemes.

Proposal 26: The accuracy and quantity of ground truth labels should be considered for ground truth label based model monitoring

Proposal 27: Further study how to use sensors' information to assist model monitoring.

Proposal 28: Further study the model monitoring schemes for AI/ML assisted positioning.

Proposal 29: Testing AI/ML model with another different drop could be a starting point for the evaluation of model monitoring methods.

Proposal 30: For the assessment of model monitoring, at least feasibility and performance should be considered as a starting point.

Proposal 31: Further study the impact of CIR truncation on positioning performance and overhead reduction for direct AI/ML positioning, at least for the following aspects:

- **The minimal length of truncated CIR to reach a target positioning accuracy.**
- **The position of truncated CIR, such as the front part or the middle part.**

Proposal 32: Further study the impact of the number of TRPs on positioning accuracy and identify the minimal number of TRPs used for positioning to ensure the target positioning accuracy for direct AI/ML positioning.

Proposal 33: For direct AI/ML positioning, support to report CIR/PDP from UE side to NW side.

Proposal 34: The minimal number of TRPs used for positioning should be identified to reach the target positioning accuracy for AI/ML assisted positioning.

Proposal 35: Further study CIR/PDP compression to support CIR/PDP reporting.

References

[1] 3GPP RP-213599, "New SI: Study on Artificial Intelligence (AI)/Machine Learning (ML) for NR Air Interface", #94e, Dec.6-17, 2021.

[2] 3GPP "Session notes for 9.2 (Study on AI/ ML for NR air interface)" TSG RAN WG1 #109-e, May 9-15, 2022.

[3] vivo, "Dataset for AI Positioning", https://commonbox.vivo.xyz/s/mEhdaRLy4Xo.

[4] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929, 2021.

[5] Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." International conference on machine learning. PMLR, 2017.

[6] Studer, Christoph, et al. "Channel charting: Locating users within the radio environment using channel state information." *IEEE Access* 6 (2018): 47682-47698.

[7] Käding, Christoph, et al. "Fine-tuning deep neural networks in continuous learning scenarios." *Asian Conference on Computer Vision*. Springer, Cham, 2016.

[8] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

[9] vivo, R1-2104363, "Discussion on potential enhancements for multipath/NLOS mitigation", TSG RAN WG1 #105-e, e-Meeting, May 10th – 27th, 2021.

[10] 3GPP, R1-145389, "Proposal on SRS estimation error modeling", TSG RAN WG1 #79, San Francisco, USA, Nov. 2014.

[11] Gao, J., M. G. Petovello, and M. E. Cannon. "Development of precise GPS/INS/wheel speed sensor/yaw rate sensor integrated vehicular positioning system." Proceedings of the 2006 national technical meeting of the institute of navigation. 2006.

[12] Chan, Yiu Tong, and K. C. Ho. "A simple and efficient estimator for hyperbolic location." IEEE transactions on signal processing 42.8 (1994): 1905-1915.

[13] 3GPP, TS38.305, V17.3.0 , 2022-12.

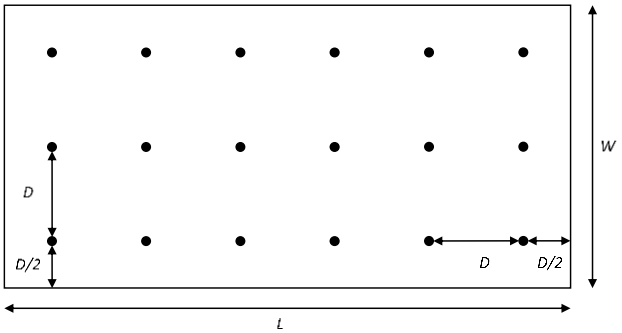
[14] vivo, R1-2302482, "Other aspects on AIML for positioning accuracy enhancement", TSG RAN WG1 #112bis-e, April 17th- April 26th, 2023.

[15] CAICT , Huawei, vivo, "2022WAIC AI/ML based High-Precision Positioning", https://www.datafountain.cn/competitions/575/ranking?isRedance=0&sch=1954

Appendix A

The common simulation parameters of scenarios can be found in Table A.1.

Table A.1 Parameters of InF scenario(s)

Parameter	Values
Scenario	InF-HH (High Tx, High Rx), InF-SL (Sparse-clutter, Low BS), InF-DL (Dense-clutter, Low BS), InF-SH (Sparse-clutter, High BS), InF-DH (Dense-clutter, High BS) – Note 1
Hall size	InF-HH: 300x150 m InF-SL: 120x60 m InF-DL: 300x150 m InF-SH: 300x150 m InF-DH: 120x60 m
Room height	10 m
gNB antenna configuration	(M, N, P, Mg, Ng) = (4, 4, 2, 1, 1), dH=dV=0.5λ for FR1.
UE antenna configuration	(M, N, P, Mg, Ng) = (1, 2, 2, 1, 1). dH=0.5λ for FR1.
Penetration loss	0dB
Number of floors	1
UE horizontal drop procedure	Uniformly distributed over the horizontal evaluation area for obtaining the CDF values for positioning accuracy. The evaluation area should be at least the convex hull of the horizontal BS deployment. It can also be the whole hall area if the CDF values for positioning accuracy is obtained from whole hall area.
BS deployment	18 BSs on a square lattice with spacing D, located D/2 from the walls. - for the small hall (L=120m x W=60m): D=20m - for the big hall (L=300m x W=150m): D=50m 
UE distribution	uniform dropping for indoor with minimum 2D distance of 0 m
UE antenna height	1.5m
gNB antenna height	BS height = 1.5 m for InF-SL and InF-DL BS height = 8 m for for InF-SH, InF-DH and InF-HH
Carrier frequency	3.5G Hz
Bandwidth	100M Hz
Clutter density: r	Low clutter density: 20% High clutter density: 60%
Clutter height: h_c	Low clutter density: 2 m High clutter density: 6 m
Clutter size: $d_{clutter}$	Low clutter density: 10 m High clutter density: 2 m
Note 1: According to Table A.2.1-7 in 3GPP TR 38.802	

Appendix B

The simulation parameters related to AI model training can be found in Table B.1.

Table B.1 Parameters of AI model training

Parameter	Model 1	Model 2
methods	Direct AI/ML positioning	AI/ML assisted positioning
AI model	Vision Transformer	FNN
BS number	18	18
CIR length	256	4096
Input	CIR 256x1x18	CIR 4096x1
Output	Location (x, y)	TOA
Synchronization	Ideal	Ideal
Channel estimation	Ideal	Ideal
Learning rate	0.002	0.002
Batch size	100	100
Epoch	1k	1k
Loss function	Mean absolute error	Mean absolute error
Optimizer	Adam	Adam
Training dataset	25k	25k
Validation dataset	1k	1k
Test dataset	1k	1k
Framework for finetuning	/	Model agnostic meta learning

The Vision Transformer evolves from typical Transformer model widely used in natural language processing, and consists of an encoder of typical Transformer model and an additional Embedding block.

Appendix C

RAN1 #109e

At the RAN1 #109e meeting, some agreements on simulation assumption and reporting KPI have been reached, which are listed as follows:

Agreement

The IIoT indoor factory (InF) scenario is a prioritized scenario for evaluation of AI/ML based positioning.

Agreement

For evaluation of AI/ML based positioning, at least the InF-DH sub-scenario is prioritized in the InF deployment scenario for FR1 and FR2.

Agreement

For InF-DH channel, the prioritized clutter parameters {density, height, size} are:

- {60%, 6m, 2m};
- {40%, 2m, 2m};

Note: an individual company may treat {40%, 2m, 2m} as optional in their evaluation considering their specific AI/ML design.

Agreement

For evaluation of AI/ML based positioning, reuse the common scenario parameters defined in Table 6-1 of TR 38.857.

Agreement

For evaluation of InF-DH scenario, the parameters are modified from TR 38.857 Table 6.1-1 as shown in the table below.

- The parameters in the table are applicable to InF-DH at least. If another InF sub-scenario is prioritized in addition to InF-DH, some parameters in the table below may be updated.

Agreement

For AI/ML-based positioning evaluation, the baseline performance to compare against is that of existing Rel-16/Rel-17 positioning methods.

- As a starting point, each participating company report the specific existing positioning method (e.g., DL-TDOA, Multi-RTT) used as comparison.

Agreement

For all scenarios and use cases, the main KPI is the CDF percentiles of horizonal accuracy.

- Companies can optionally report vertical accuracy.

Agreement

The CDF percentiles to analyse are: {50%, 67%, 80%, 90%}.

- 90% is the baseline. {50%, 67% 80%} are optional.

Agreement

Target positioning requirements for horizonal accuracy and vertical accuracy are not defined for AI/ML-based positioning evaluation.

Agreement

For evaluation of AI/ML based positioning, the KPI includes the model complexity and computational complexity.

- FFS: the details of model complexity and computational complexity

Agreement

Synthetic dataset generated according to the statistical channel models in TR38.901 is used for model training, validation, and testing.

Agreement

The dataset is generated by a system level simulator based on 3GPP simulation methodology.

Agreement

As a starting point, the training, validation and testing dataset are from the same large-scale and small-scale propagation parameters setting. Subsequent evaluation can study the performance when the training dataset and testing dataset are from different settings.

Agreement

For AI/ML-based positioning evaluation, RAN1 does not attempt to define any common AI/ML model as a baseline.

Agreement

The entry "UE horizontal drop procedure" in the simulation parameter table for InF is updated to the following.

UE horizontal drop procedure	Uniformly distributed over the horizontal evaluation area for obtaining the CDF val positioning accuracy, The evaluation area should be selected from - (baseline) the whole hall area, and the CDF values for positioning accuracy is o from whole hall area. - (optional) the convex hull of the horizontal BS deployment, and the CDF val positioning accuracy is obtained from the convex hull.

Agreement

The entries "UE antenna height" and "gNB antenna height" in the simulation parameter table for InF is updated to the following.

UE antenna height	Baseline: 1.5m (Optional): uniformly distributed within [0.5, X2]m, where X2 = 2m for scenario 1(InF-SH) and X2=h_c for scenario 2 (InF-DH)	
…	…	
gNB antenna height	Baseline: 8m (Optional): two fixed heights, either {4, 8} m, or {max(4,h_c), 8}.	

Agreement

If spatial consistency is enabled for the evaluation, companies model at least one of: large scale parameters, small scale parameters and absolute time of arrival, where

- the large scale parameters are according to Section 7.5 of TR 38.901 and correlation distance = $d_{clutter}/2$ for InF (Section 7.6.3.1 of TR 38.901)
- the small scale parameters are according to Section 7.6.3.1 of TR 38.901
- the absolute time of arrival is according to Section 7.6.9 of TR 38.901

Agreement

If spatial consistency is enabled for the evaluation of AI/ML based positioning, the baseline evaluation does not incorporate spatially consistent UT/BS mobility modelling (Section 7.6.3.2 of TR 38.901).

- It is optional to implement spatially consistent UT/BS mobility modelling (Section 7.6.3.2 of TR 38.901).

Agreement

For evaluation of AI/ML based positioning, companies are encouraged to evaluate the model generalization.

- FFS: the metrics for evaluating the model generalization (e.g., model performance based on agreed KPIs under different settings)

Agreement

Companies are encouraged to provide evaluation results for:

- Direct AI/ML positioning
  - Companies are encouraged to describe at least the following implementation details for the evaluation
    - details of the channel observation used as the input of the AI/ML model inference (e.g., type and size of model input), model input acquisition and pre-processing
- AI/ML assisted positioning
  - Companies are encouraged to describe at least the following implementation details for the evaluation
    - details of the channel observation used as the input of the AI/ML model inference (e.g., type and size of model input), model input acquisition and pre-processing
    - details of the output of the AI/ML model inference, how the AI/ML model output is used to obtain the UE's location

Agreement

When reporting evaluation results with direct AI/ML positioning and/or AI/ML assisted positioning, proponent company is expected to describe if a one-sided model or a two-sided model is used.

- If one-sided model (i.e., UE-side model or network-side model), the proponent company report which side the model inference is performed (e.g. UE, network), and any details specific to the side that performs the AI/ML model inference.
- If two-sided model, the proponent company report which side (e.g., UE, network) performs the first part of interference, and which side (e.g., network, UE) performs the remaining part of the inference.

Agreement

For evaluation of AI/ML based positioning, the computational complexity can be reported via the metric of floating point operations (FLOPs).

- Note: For AI/ML assisted methods, computational complexity for the AI/ML model is only one component of the overall complexity for estimating the UE's location.
- Note: Other metrics to measure the computational complexity are not precluded.

Agreement

For evaluation of AI/ML based positioning, details of the training dataset generation are to be reported by proponent company. The report may include (in addition to other selected settings, if applicable):

- The size of training dataset, for example, the total number of UEs in the evaluation area for generating training dataset;
- The distribution of UE location for generating the training dataset may be one of the following:
    - Option 1: grid distribution, i.e., one training data is collected at the center of one small square grid, where, for example, the width of the square grid can be 0.25/0.5/1.0 m.

Option 2: uniform distribution, i.e., the UE location is randomly and uniformly distributed in the evaluation area.

RAN1 #110

At the RAN1 #110 meeting, some agreements on simulation assumption, KPI and futher research directions have been reached, which are listed as follows:

Agreement

For AI/ML-based positioning, both approaches below are studied and evaluated by RAN1:

- Direct AI/ML positioning
- AI/ML assisted positioning

Agreement

For AI/ML-based positioning, study impact from implementation imperfections.

Agreement

For evaluation of AI/ML based positioning, the model complexity is reported via the metric of "number of model parameters".

Agreement

To investigate the model generalization capability, at least the following aspect(s) are considered for the evaluation for AI/ML based positioning:

b) Different drops

- Training dataset from drops {A0, A1,…, AN-1}, test dataset from unseen drop(s) (i.e., different drop(s) than any in {A0, A1,…, AN-1}). Here N>=1.

c) Clutter parameters, e.g., training dataset from one clutter parameter (e.g., {40%, 2m, 2m}), test dataset from a different clutter parameter (e.g., {60%, 6m, 2m});

d) Network synchronization error, e.g., training dataset without network synchronization error, test dataset with network synchronization error;

Other aspects are not excluded.

Note: It's up to participating companies to decide whether to evaluate one aspect at a time, or evaluate multiple aspects at the same time.

Agreement

When providing evaluation results for AI/ML based positioning, participating companies are expected to describe data labelling details, including:

- Meaning of the label (e.g., UE coordinates; binary identifier of LOS/NLOS; ToA)

- Percentage of training data without label, if incomplete labeling is considered in the evaluation
- Imperfection of the ground truth labels, if any

Agreement

For evaluation of AI/ML based positioning, study the performance impact from availability of the ground truth labels (i.e., some training data may not have ground truth labels). The learning algorithm (e.g., supervised learning, semi-supervised learning, unsupervised learning) is reported by participating companies.

Agreement

For AI/ML-based positioning, for evaluation of the potential performance benefits of model finetuning, report at least the following:

- training dataset setting (e.g., training dataset size necessary for performing model finetuning)
- horizontal positioning accuracy (in meters) before and after model finetuning.

Agreement

For both direct AI/ML positioning and AI/ML assisted positioning, the following table is adopted for reporting the evaluation results.

Table *X*. Evaluation results for AI/ML model deployed on [UE or network]-side, [with or without] model generalization, [short model description]

Model input	**Model output**	**Label**	**Clutter param**	**Dataset size**		**AI/ML complexity**		**Horizontal positioning accuracy at CDF=90% (meters)**
				Training	test	Model complexity	Computational complexity	AI/ML

To report the following in table caption:

- Which side the model is deployed
- Model generalization investigation, if applied
- Short model description: e.g., CNN

Further info for the columns:

- Model input: input type and size
- Model output: output type and size
- Label: meaning of ground truth label; percentage of training data set without label if data labeling issue is investigated (default = 0%)
- Clutter parameter: e.g., {60%, 6m, 2m}
- Dataset size, both the size of training/validation dataset and the size of test dataset
- AI/ML complexity: both model complexity in terms of "number of model parameters", and computational complexity in terms of FLOPs
- Horizontal positioning accuracy: the accuracy (in meters) of the AI/ML based method
- Note: To report other simulation assumptions, if any.

Offline Agreement

For evaluation of AI/ML assisted positioning, an intermediate performance metric of model output is reported.

- FFS: Detailed definition of the intermediate performance metric of the model output

Offline Agreement

To investigate the model generalization capability, the following aspect is also considered for the evaluation of AI/ML based positioning:

e) UE/gNB RX and TX timing error.

- The baseline non-AI/ML method may enable the Rel-17 enhancement features (e.g., UE Rx TEG, UE RxTx TEG).

RAN1 #110b-e

At the RAN1 #110b-e meeting, some agreements on simulation assumption and reporting KPI have been reached, which are listed as follows:

Agreement

To investigate the model generalization capability, the following aspect is also considered for the evaluation of AI/ML based positioning:

(e) InF scenarios, e.g., training dataset from one InF scenario (e.g., InF-DH), test dataset from a different InF scenario (e.g., InF-HH)

Agreement

For both direct AI/ML positioning and AI/ML assisted positioning, if fine-tuning is not evaluated, the template agreed in RAN1#110 is updated to the following for reporting the evaluation results.

Table X. Evaluation results for AI/ML model deployed on [UE or network]-side, [short model description]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Test	Train	test	Model complexity	Computation complexity	AI/ML

Agreement

For both direct AI/ML positioning and AI/ML assisted positioning, if fine-tuning is evaluated, the template agreed in RAN1#110 is updated to the following for reporting the evaluation results.

Table X. Evaluation results for AI/ML model deployed on [UE or network]-side, [short model description]

Model input	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**			**Dataset size**			**AI/ML complexity**		**Horizontal pos. accuracy at CDF=90% (m)**
			Train	Fine-tune	Test	Train	Fine-tune	test	Model complexity	Computation complexity	AI/ML

Agreement

For AI/ML-assisted positioning, companies report which construction is applied in their evaluation:

(a) Single-TRP construction: the input of the ML model is the channel measurement between the target UE and a single TRP, and the output of the ML model is for the same pair of UE and TRP.
(b) Multi-TRP construction: the input of the ML model contains N sets of channel measurements between the target UE and N (N>1) TRPs, and the output of the ML model contains N sets of values, one for each of the N TRPs.

Note: For a measurement (e.g., RSTD) which is a relative value between a given TRP and a reference TRP, the TRP in “single-TRP” and “multi-TRP” refers to the given TRP only.

Note: For single-TRP construction, companies report whether they consider same model for all TRPs or N different models for TRPs

Conclusion

For evaluation of AI/ML based positioning, suspend the discussion on intra-site (or zone-specific) variations until concepts and channel model construction not in TR38.901 (e.g., "intra-site" or "zone") are clarified under AI 9.2.1.

- Note: An individual company can still submit evaluation results for intra-site variation.

Conclusion

For evaluation of AI/ML based positioning, the sampling period is selected by proponent companies. Each company report the sampling period used in their evaluation.

Agreement

For evaluation of AI/ML assisted positioning, the following intermediate performance metrics are used:

- LOS classification accuracy, if the model output includes LOS/NLOS indicator of hard values, where the LOS/NLOS indicator is generated for a link between UE and TRP;
- Timing estimation accuracy (expressed in meters), if the model output includes timing estimation (e.g., ToA, RSTD).
- Angle estimation accuracy (in degrees), if the model output includes angle estimation (e.g., AoA, AoD).
- Companies provide info on how LOS classification accuracy and timing/angle estimation accuracy are estimated, if the ML output is a soft value that represents a probability distribution (e.g., probability of LOS, probability of timing, probability of angle, mean and variance of timing/angle, etc.)

Conclusion

For evaluation of AI/ML based positioning, it's up to each company to take into account the channel estimation error in their evaluation. Companies describe the details of their simulation assumption, e.g., realistic or ideal channel estimation, error models, receiver algorithms.

Agreement

For AI/ML assisted positioning, when single-TRP construction is used for the AI/ML model, companies report at least the AI/ML complexity (Model complexity, Computation complexity) for N TRPs, which are used to determine the position of a target UE.

Table. Model complexity and computation complexity to support N TRPs for a target UE

	Model complexity to support N TRPs	**Computation complexity to process N TRPs**
Single-TRP, same model for N TRPs	$\boldsymbol{P_S}$ When the model is at UE-side, where $\boldsymbol{P_S}$ is the model complexity for the same model. FFS: if the model is at network-side	$\boldsymbol{N \times C_S}$ Where $\boldsymbol{C_S}$ is the computation complexity of the same model for one TRP.
Single-TRP, N models for N TRPs	When the model is at UE-side, $\sum_{i=1,\ldots N} \boldsymbol{P_{S,i}}$ Where $\boldsymbol{P_{S,i}}$ is the model complexity for the i-th AI/ML model. FFS: if the model is at network-side	$\sum_{i=1,\ldots N} \boldsymbol{C_{S,i}}$ Where $\boldsymbol{C_{S,i}}$ is the computation complexity for the i-th AI/ML model.

Multi-TRP (i.e., one model for N TRPs)	$\boldsymbol{P_M}$ Where $\boldsymbol{P_M}$ is the model complexity for the one model.	$\boldsymbol{C_M}$ Where $\boldsymbol{C_M}$ is the computation complexity for the one model.

Agreement

For AI/ML based positioning, if an InF scenario different from InF-DH is evaluated for the model generalization capability, the selected parameters (e.g., clutter parameters) are compliant with TR 38.901 Table 7.2-4 (Evaluation parameters for InF).

- Note: In TR 38.857 Table 6.1-1 (Parameters common to InF scenarios), InF-SH scenario uses the clutter parameter {20%, 2m, 10m} which is compliant with TR 38.901.

Agreement

For the model input used in evalutions of AI/ML based positioning, if time-domain channel impulse response (CIR) or power delay profile (PDP) is used as model input in the evaluation, companies report the input dimension N_{TRP} * N_{port} * N_t, where N_{TRP} is the number of TRPs, N_{port} is the number of transmit/receive antenna port pairs, N_t is the number of time domain samples.

- Note: CIR and PDP may have different dimensions.

Note: Companies provide details on their assumption on how PDP is constructed and how (if applicable) it is mapped to N_t samples.

RAN1 #111

At the RAN1 #111 meeting, some agreements on simulation assumption and reporting KPI have been reached, which are listed as follows:

Agreement

Study how AI/ML positioning accuracy is affected by: user density/size of the training dataset.

Note: details of user density/size of training dataset to be reported in the evaluation.

Agreement

For reporting the model input dimension N_{TRP} * N_{port} * N_t of CIR and PDP, N_t refers to ~~the first~~ N_t consecutive time domain samples.

- If N'_t ($N'_t < N_t$) samples with the strongest power are selected as model input, with remaining ($N_t - N'_t$) time domain samples set to zero, then companies report value N'_t in addition to N_t. It is also assumed that timing info for the N'_t samples need to be provided as model input.

Agreement

For reporting the model input dimension N_{TRP} * N_{port} * N_t:

- If the model input is CIR, then each input value of CIR is a complex number, i.e. it contains two real values, either {real, imaginary} or {magnitude, phase}.
- If the model input is PDP, then each input value of PDP is a real value.

Agreement

At least for model inference of AI/ML assisted positioning, evaluate and report the AI/ML model output, including (a) the type of information (e.g., ToA, RSTD, AoD, AoA, LOS/NLOS indicator) to use as model output, (b) soft information vs hard information, (c) whether the model output can reuse existing measurement report (e.g., NRPPa, LPP).

Agreement

For AI/ML assisted positioning, evaluate the three constructions:

- Single-TRP, same model for N TRPs
- Single-TRP, N models for N TRPs
- Multi-TRP (i.e., one model for N TRPs)

Note: Individual company may evaluate one or more of the three constructions.

Agreement

For AI/ML assisted approach, study the performance of model monitoring metrics at least where the metrics are obtained from inference accuracy of model output.

Agreement

For both direct and AI/ML assisted positioning methods, investigate at least the impact of the amount of fine-tuning data on the positioning accuracy of the fine-tuned model.

- The fine-tuning data is the training dataset from the target deployment scenario.

Agreement

For the RAN1#110bis agreement on the calculation of model complexity, the FFS are resolved with the following update:

	Model complexity to support N TRPs
Single-TRP, same model for N TRPs	P_S where P_S is the model complexity for one TRP and the same model is used for N TRPs.
Single-TRP, N models for N TRPs	$\sum_{i=1,\ldots N} P_{S,i}$ Where $P_{S,i}$ is the model complexity for the i-th AI/ML model.

Note: The reported model complexity above is intended for inference and may not be directly applicable to complexity of other LCM aspects.

Observation

Direct AI/ML positioning can significantly improve the positioning accuracy compared to existing RAT-dependent positioning methods when the generalization aspects are not considered.

- For InF-DH with clutter parameter setting {60%, 6m, 2m}, evaluation results submitted to RAN1#111 indicate that the direct AI/ML positioning can achieve horizontal positioning accuracy of <1m at CDF=90%, as compared to >15m for conventional positioning method.

Agreement

For AI/ML based positioning, company optionally evaluate the impact of at least the following issues related to measurements on the positioning accuracy of the AI/ML model. The simulation assumptions reflecting these issues are up to companies.

- SNR mismatch (i.e., SNR when training data are collected is different from SNR when model inference is performed).
- Time varying changes (e.g., mobility of clutter objects in the environment)
- Channel estimation error

Conclusion

Companies describe how their computational complexity values are obtained.

- It is out of 3GPP scope to consider computational complexity values that have platform-dependency and/or use implementation (hardware and software) optimization solutions.

Observation

AI/ML assisted positioning can significantly improve the positioning accuracy compared to existing RAT-dependent positioning methods when the generalization aspects are not considered.

- For InF-DH with clutter parameter setting {40%, 2m, 2m}, evaluation results submitted to RAN1#111 indicate that the AI/ML assisted positioning can achieve horizontal positioning accuracy of <0.4m at CDF=90%, as compared to >9m for conventional positioning method.
- For InF-DH with clutter parameter setting {60%, 6m, 2m}, evaluation results submitted to RAN1#111 indicate that the AI/ML assisted positioning can achieve horizontal positioning accuracy of <1m at

CDF=90%, as compared to >15m for conventional positioning method.
Note: how to capture the observation(s) into TR is separate discussion.

Agreement

For AI/ML assisted approach, for a given AI/ML model design (e.g., input, output, single-TRP vs multi-TRP), identify the generalization aspects where model fine-tuning/mixed training dataset/model switching is necessary.

RAN1 #112

At the RAN1 #112 meeting, some agreements on simulation assumption and reporting KPI have been reached, which are listed as follows:

Agreement

For both direct AI/ML positioning and AI/ML assisted positioning, companies include the evaluation area in their reporting template, assuming the same evaluation area is used for training dataset and test dataset.

Note:

- Baseline evaluation area for InF-DH = 120x60 m.
- if different evaluation areas are used for training dataset and test dataset, they are marked out separately under "Train" and "Test" instead.

Table X. Evaluation results for AI/ML model deployed on [UE or network]-side, [with or without] model generalization, [short model description], UE distribution area = [e.g., 120x60 m, 100x40 m]

Model input	**Model output**	**Label**	**Clutter param**	**Dataset size**		**AI/ML complexity**		**Horizontal p accuracy at CDF=90%**
				Train	Test	Model complexity	Computation complexity	AI/ML

Table X. Evaluation results for AI/ML model deployed on [UE or network]-side, [short model description], UE distribution area = [e.g., 120x60 m, 100x40 m]

lodel put	**Model output**	**Label**	**Settings (e.g., drops, clutter param, mix)**		**Dataset size**		**AI/ML complexity**		**Horizontal po accuracy CDF=90% (m**
			Train	Test	Train	Test	Model complexity	Computation complexity	AI/ML

Agreement

The agreement made in RAN1#110 AI 9.2.4.1 is updated by adding additional note:

Note: if complex value is used in modelling process, the number of the model parameters is doubled, which is also applicable for other AIs of AI/ML

Agreement

For both the direct AI/ML positioning and AI/ML assisted positioning, study the model input, considering the tradeoff among model performance, model complexity and computational complexity.

- The type of information to use as model input. The candidates include at least: time-domain CIR, PDP.
- The dimension of model input in terms of N_{TRP}, N_t, and N_t'.
- Note: For the direct AI/ML positioning, model input size has impact to signaling overhead for model inference.

Agreement

For **direct** AI/ML positioning, study the performance of model monitoring methods, including:

- Label based methods, where ground truth label (or its approximation) is provided for monitoring the accuracy of model output.
- Label-free methods, where model monitoring does not require ground truth label (or its approximation).

Agreement

For AI/ML **assisted** approach, study the performance of label-free model monitoring methods, which do not require ground truth label (or its approximation) for model monitoring.

Conclusion

- No dedicated evaluation is needed for the positioning accuracy performance of model switching
- It does not preclude future discussion on model switching related performance

Agreement

For direct AI/ML positioning, study the impact of labelling error to positioning accuracy

- The ground truth label error in each dimension of x-axis and y-axis can be modeled as a truncated Gaussian distribution with zero mean and standard deviation of L meters, with truncation of the distribution to the [-2*L, 2*L] range.
    - Value L is up to sources.
- Other models are not precluded
- [Whether/how to study the impact of labelling error to label-based model monitoring methods]
- [Whether/how to study the impact of labelling error for AI/ML assisted positioning.]

Observation

Evaluation of the following generalization aspects show that the positioning accuracy of direct AI/ML positioning deteriorates when the AI/ML model is trained with dataset of one deployment scenario, while tested with dataset of a different deployment scenario.

- The generalization aspects include:
    - Different drops
    - Different clutter parameters
    - Different InF scenarios
    - Network synchronization error
- Companies have provided evaluation results which show that the positioning accuracy on the test dataset can be improved by better training dataset construction and/or model fine-tuning/re-training.
    - Better training dataset construction: The training dataset is composed of data from multiple deployment scenarios, which include data from the same deployment scenario as the test dataset.
    - Model fine-tuning/re-training: the model is re-trained/fine-tuned with a dataset from the same deployment scenario as the test dataset.

Note: ideal model training and switching may provide the upper bound of achievable performance when the AI/ML model needs to handle different deployment scenarios.