VERITAS: Verifying the Performance of AI-native Transceiver Actions in Base-Stations

Nasim Soltani[§], Michael Loehning[†], Kaushik Chowdhury[§]

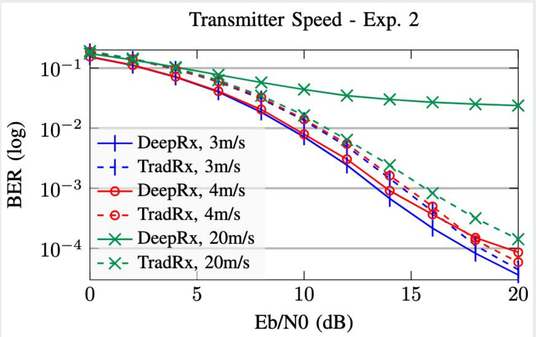
[§]WNCG at The University of Texas at Austin, [†]National Instruments - Test and Measurement Group of Emerson, Dresden, Germany

Personal Website

Motivation

AI-native Rx **outperforms** traditional Rx (TradRx) as long as it is deployed in similar environments (channel conditions) that it is trained under.

AI-native Rx performance is not guaranteed in **all** channel conditions. In some training / deployment configurations, AI-native receiver might face **performance drop** compared to the TradRx.



We study changes in 3 different channel parameters between the training and deployment datasets.

- Channel profile (LOS/NLOS)
- Transmitter speed (m/s)
- Delay Spread (ns)

And find performance drop cases as the following:

		Channel Profile Exp. 2	Tx Speed Exp. 2	Delay Spread Exp. 2
Training	Channel Profile	tdl_d	tdl_d	tdl_b
	Tx Speed (m/s)	10	0, 1, 2	2
	Delay Spread (ns)	400	400	10, 50, 80
Test	Channel Profile	tdl_a, tdl_b, tdl_c	tdl_d	tdl_b
	Tx Speed (m/s)	10	≥ 4	2
	Delay Spread (ns)	400	400	> 200

Research Question

Is there a method for verifying the performance of AI-native receiver? Is there a way to **identify if the AI-native receiver is underperforming the TradRx during deployment** and initiate a retraining process to adapt the AI-native receiver to the new channel conditions?

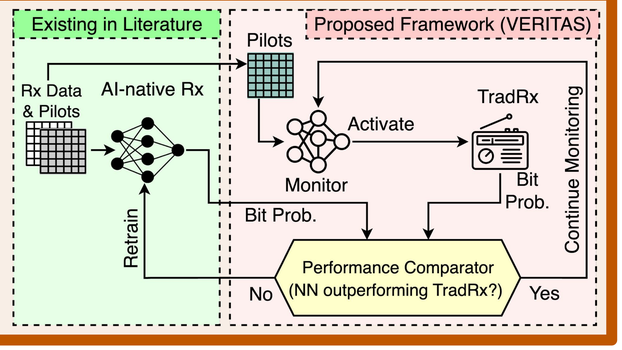
VERITAS: For Verifying the Performance of AI-native Rx

Two main components:

(a) Monitor (b) Performance Comparator

The Monitor continuously scans the wireless channel, and as soon as it detects a change, it activates TradRx.

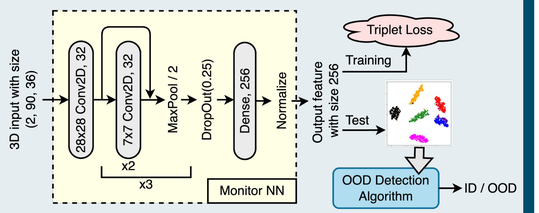
Bit probabilities of AI-native Rx and TradRx are compared to find the underperforming receiver. If necessary, the AI-native Rx is updated through retraining to adapt to the new wireless channel.



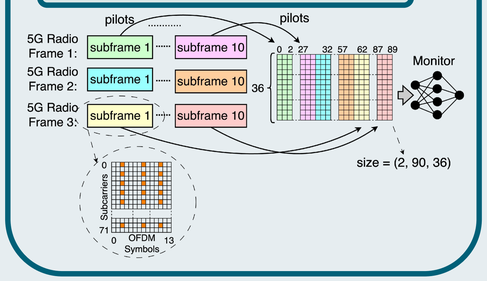
(a) The Monitor

The Monitor has two main components:

- An NN-based feature extractor trained with triplet loss.
- A non-parametric and distance-based out-of-distribution (OOD) detection algorithm.



1 Monitor NN Input Preparation



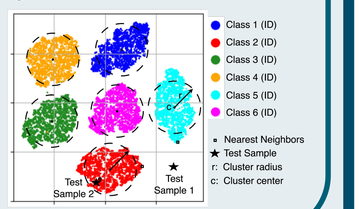
2 OOD Detection Algorithm

Post-training - Characterizing ID clusters:

- Center
- Radius

During Deployment:

- Return K-nearest Neighbors
- Detect OOD samples



(b) The Performance Comparator

The Performance Comparator has two steps:

1

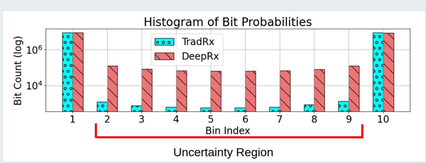
Log Likelihood Ratios (LLRs) out of TradRx and DeepRx are converted to bit Probabilities.

$$P = \frac{1}{1 + e^{\mathrm{LLR}}}$$

2

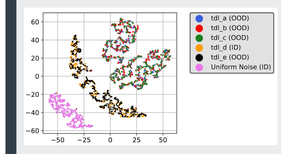
Histograms of bit probabilities for TradRx and DeepRx are created and compared to find the underperforming receiver.

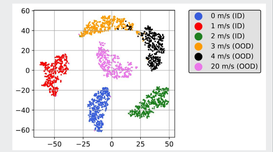
$$\mathrm{bin}_b : \begin{cases} [\frac{b-1}{10}, \frac{b}{10}) & b = 1, 2, 3, ..., 9 \\ [\frac{b-1}{10}, \frac{b}{10}] & b = 10 \end{cases}$$



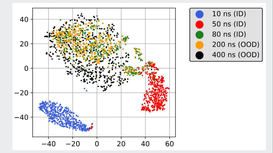
Results

(a) Monitor - Clusters

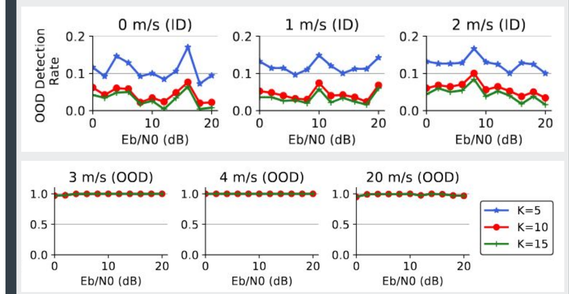




t-SNE plots show good cluster separation between the ID and OOD classes.

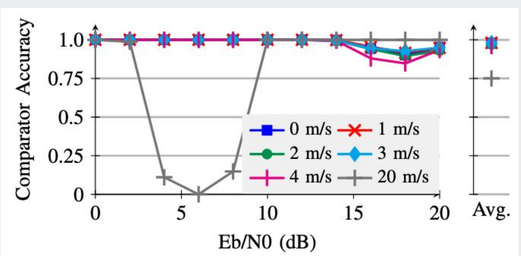


(a) Monitor – OOD Detection Rate



- **Low False Positives**: ID classes show low OOD detection rate across different Eb/N0s.
- **High True Positives**: OOD classes show high OOD detection rate across different Eb/N0s.

(b) Performance Comparator - Accuracy



Performance Comparator achieves high accuracy in **identifying the underperforming receiver,** for ID and OOD classes, and initiates AI-native Rx retraining, if necessary.

N. Soltani, M. Loehning, and K. Chowdhury, "VERITAS: Verifying the Performance of AI-native Transceiver Actions in Base-Stations," Under Review at IEEE Journal, available on ArXiv:2501.09761 (2025).