Applying NLOS Classification and Error Correction Techniques to UWB Systems: Lessons Learned and Recommendations

Michael Stocker*
michael.stocker@tugraz.at
Institute of Technical Informatics
Graz University of Technology
Graz, Austria

Markus Gallacher*
markus.gallacher@tugraz.at
Institute of Technical Informatics
Graz University of Technology
Graz, Austria

Carlo Alberto Boano
cboano@tugraz.at
Institute of Technical Informatics
Graz University of Technology
Graz, Austria

Kay Römer
roemer@tugraz.at
Institute of Technical Informatics
Graz University of Technology
Graz, Austria

ABSTRACT

In recent years, research on the detection and mitigation of non-line-of-sight (NLOS) conditions in the context of ultra-wideband ranging has received increasing attention. As a result, numerous statistical and machine learning methods have been proposed, and a selection of datasets has been made available to the community. In an attempt to benchmark the performance of state-of-the-art NLOS classification and error correction techniques on a newly-built ultra-wideband testbed at our premises, we have observed how reusing publicly-available datasets and applying existing solutions is a complex and error-prone task. Indeed, a multitude of minor details in the selection, pre-processing, collection, labeling, and blending of datasets can have a profound impact on the correctness of the employed methods and on the achieved performance. In this paper, we summarize the lessons we have learned, pointing out potential pitfalls and distilling a few recommendations for researchers and practitioners approaching this research domain.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; • **Networks** → **Location based services**.

KEYWORDS

DW1000, eXtreme Gradient Boosting, Machine Learning, NLOS, Support Vector Machines, Testbed, Ultra-Wideband, Wireless.

ACM Reference Format:
Michael Stocker, Markus Gallacher, Carlo Alberto Boano, and Kay Römer. 2023. Applying NLOS Classification and Error Correction Techniques to UWB Systems: Lessons Learned and Recommendations. In *Cyber-Physical Systems and Internet of Things Week 2023 (CPS-IoT Week Workshops '23), May 9–12, 2023, San Antonio, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3576914.3587522

1 INTRODUCTION

Ultra-wideband (UWB) has recently become the technology of choice to develop highly-accurate indoor localisation systems. Due to its excellent temporal resolution and resilience to multi-path interference, UWB technology allows to obtain cm-level accurate distance measurements. The popularity and ubiquity of UWB devices is set to increase in the coming years, as big industry players such as Apple, Samsung, BMW, and Volkswagen, have started integrating UWB transceivers into their smartphones and cars. Moreover, several system providers have emerged, (e.g., Ubisense, Kinexon, and Sewio), fueling the adoption of UWB services, and paving the way for applications such as asset tracking.

Poor performance in NLOS conditions. Unfortunately, the precision and accuracy of UWB rangings degenerate when obstacles partially or fully occlude the direct path between two devices. These *non-line-of-sight (NLOS)* conditions hinder UWB transceivers from accurately measuring the time-of-arrival (ToA) of packets, which is used to estimate the distance between devices in the two-way ranging (TWR) process. Detecting the presence of NLOS conditions and correcting their impact on ranging measurements is hence important, and has triggered a large body of research work. Several solutions use statistical methods [3, 14], such as using the variance of consecutive ranging measurements for NLOS classification. Other approaches [6, 9] use channel statistics extracted from the channel impulse response (CIR) of received packets. Due to the complex interplay between obstacles, the surrounding environment, and its impact on the CIR, there has been an increased interest in using machine learning (ML) models for NLOS classification and error correction. Maranò et al. [7, 13] were among the first to use channel statistics extracted from the CIR as features to train support vector machines (SVMs) for NLOS classification and for correcting the ranging errors introduced by NLOS conditions. More recently, a plethora of works [1, 4, 12] have emerged using various types of deep neural networks for NLOS error classification and correction.

Applicability and performance of existing solutions. Despite this large body of work, the *performance* of the proposed solutions in different real-world settings has not been studied in detail yet. In fact, most of the existing studies tackling the NLOS problem train and evaluate the proposed ML models using self-collected data. For example, Bregar et al. [4], and Angarano et al. [1] collect data in different rooms, and use some rooms for training and other rooms for testing. Stocker et al. [11], instead, collect data in different environments and do not differentiate among them during training/testing, while making sure to never test at a location whose datapoints were used for training. This limits the generality of the results (as they

*Both authors contributed equally to this research.

have rarely been *validated by others in different settings*), and calls for studies and benchmarking initiatives to better understand the performance of existing solutions in the wild.

Also the *ease of reuse* and *applicability* of existing solutions has not been the focus of existing research. Based on our experience, reusing publicly-available datasets and applying existing solutions is a complex and error-prone task. On the one hand, no pre-trained models are publicly available, which means that they must be re-created and re-trained before deployment. On the other hand, a plethora of tiny details in the selection, pre-processing, collection, labeling, and blending of datasets can strongly affect performance.

Contributions. In this paper, we benchmark the performance of existing NLOS classification and error correction techniques trained with publicly-available datasets on our newly-built UWB testbed. Our experiments show, among others, that the achieved classification and especially the error correction performance can vary *tremendously* due to tiny details in the selection, pre-processing, collection, labeling, and blending of datasets. Despite many attempts, the achieved NLOS classification and error correction performance in our testbed environment is – even with the best configuration – often lower than the one presented by related works training and testing their solution using an own dataset. Moreover, the performance changes drastically as a function of the employed training dataset, and it does not necessarily increase when blending more datasets together. After presenting our experimental results, we summarize all the lessons we have learned during our investigation, pointing out potential pitfalls and distilling a few recommendations for researchers and practitioners approaching this research domain.

Paper outline. The paper proceeds as follows. § 2 introduces the reader to UWB technology and to the NLOS problem, illustrating relevant related works in the field. § 3 describes our attempt in benchmarking existing solutions in our testbed, and provides an overview of the used datasets, models, and obtained results. § 4 compiles a list of the most relevant insights we obtained during our benchmarking effort in form of lessons and recommendations. § 5 concludes the paper with an outlook on future work.

2 A PRIMER ON UWB AND NLOS

UWB systems utilize a bandwidth of ≥ 500 MHz, and are thus able to communicate by sending very short pulses (≈ 2ns). This results in several benefits: among them is the high temporal resolution of incoming signals, meaning that UWB receivers can precisely separate multi path components (MPCs) from the LOS component (also called first path – FP) of a received signal and precisely determine the time-of-arrival (ToA) of packets. The ToA is used to determine a packet's time-of-flight (ToF), which is proportional to the distance between transmitter and receiver. Two UWB devices can hence estimate the ToF and derive their distance by exchanging several messages, e.g., with the two-way ranging (TWR) process.

A crucial step for accurate ToA estimation is the analysis and correct identification of the FP within the CIR. The latter essentially contains information about propagation paths of the radio signals in the environment and is generated in the UWB receiver by accumulating several preamble symbols. The number of accumulated preamble symbols is reported in the preamble accumulation counter (`RXPACC` register in the popular Qorvo DW1000 radio).

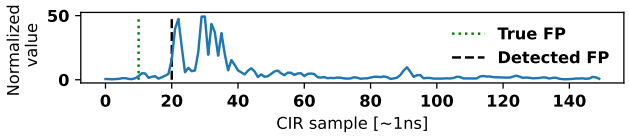

Figure 1: CIR estimate taken in NLOS conditions. The true FP peak has a lower amplitude than the following MPCs, causing the radio to erroneously mark the detected FP at a later point in time.

Fig. 1 shows an exemplary CIR estimate acquired with the DW1000 radio in the presence of NLOS conditions[1]. The true FP (dotted green line) is attenuated or blocked, and has a much lower amplitude than the following MPCs. Because of this, an MPC has falsely been identified as FP (dashed black line). The resulting wrong distance estimate can pose a serious threat to safety-critical systems such as real-time hazard detection for workers and plant operators.

Several works have proposed to use *ML methods* to lessen the effects of NLOS conditions. The intuition is that CIRs recorded in NLOS conditions follow specific patterns that can be exploited for detecting and correcting ranging errors. Among the first ones to follow this path were Maranò et al. [7], who extracted several features from the CIR (such as the rise time, the signal strength, and the delay spread), and used them to train an SVM classifier (SVC) as well as regressor (SVR) based on samples collected within a university building. Stocker et al. [11] performed similar experiments with the Qorvo `DWM1001-DEV` devices and investigated the accuracy of NLOS error correction models based on support vector regression. Barral et al. [2] used pre-computed features from Pozyx devices (based on the DW1000 radio) and compared four ML models (including SVMs and Binary Decision Trees) for NLOS classification and five models for NLOS mitigation. Angarano et al. [1] and Bregar et al. [4] used instead different types of *deep neural networks* (DNNs), which were trained on the raw CIR and automatically extracted relevant features for NLOS classification and/or regression.

3 BENCHMARKING NLOS SOLUTIONS

We benchmark the performance of existing NLOS classification and error correction techniques on a real-world testbed. After describing the experimental setup including various types of obstacles (§ 3.1), we present the ML models (§ 3.2) and public datasets for training (§ 3.3) considered in our study, and discuss our results (§ 3.4).

3.1 Testbed Facility and Experimental Setup

Hardware. We use a portion of the UWB testbed facility deployed at our university [10] consisting of 26 Qorvo `DWM1001-DEV` devices mounted on the walls across a large hallway (acting as *anchors*). We further employ additional `DWM1001-DEV` devices spread throughout the hallway (acting as mobile *tags*). Anchors and tags are depicted as orange squares and blue circles in Fig. 2, respectively.

Collected `ATA/TTA` datasets. We run two types of measurements in both LOS and NLOS conditions. First, we let pairs of anchor nodes perform double-sided TWR among each other: with a mean of 133 measurements/pair, we collect a dataset with 20606 points and refer to this as "anchor-to-anchor" (`ATA`) dataset. Second, we let the tags

[1]Note that the DW1000 radio typically arranges the CIR such that the detected FP is placed at sample index 750. We have artificially shifted it to sample index 20.

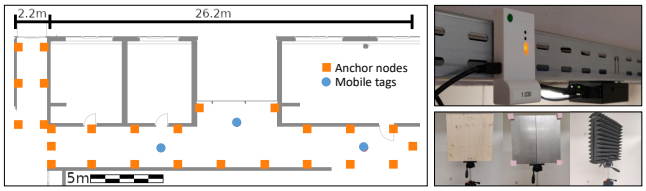

Figure 2: Overview of our experimental setup. Left: testbed topology, with 26 anchor nodes (orange squares) and 3 mobile tags (blue circles). Top-right: anchor nodes mounted on metal rail. Bottom-right: obstacles used to create NLOS conditions.

perform double-sided TWR to 13 surrounding anchors (8–10 of which in LOS, and 3–5 in NLOS). For each tag's position, we vary the tag's orientation in 45° steps: this results in a dataset with 54175 points, and we refer to this as "tag-to-anchor" (TTA) dataset. All measurements are taken using channel 5, a transmission power of 14.4 dBm, and 128 preamble symbol repetitions. We record the reported distance, the amount of accumulated preamble symbols, and 400 samples of the CIR with 64 MHz pulse repetition frequency.

Obstacles. We perform measurements within a range of 10 m. Node pairs without obstacles in between are labelled as LOS. We also place three types of obstacle (wood panel, metal steel sheet, and two pyramid foam absorbers placed back-to-back, as shown in Fig. 2) between node pairs, and label the measurements according to the obstacle type. All obstacles have a dimension of $50x50$ cm, with the wood panel and metal steel sheet being 2 cm and 2 mm thick, respectively. Obstacles are placed at 0.4 m from an anchor in the ATA dataset, and at 0.2, 0.5, and 0.8 m from the tag in the TTA dataset.

3.2 Considered ML Models

We investigate two widely-adopted ML models: SVMs and extreme gradient-boosted trees (XGBoost)[2]. While SVMs have often been used for NLOS classification and error correction [1, 2, 4, 7, 11], the use of XGBoost trees is relatively new in this field. XGBoost trees use an ensemble of small decision trees that learn from the residual errors of the previous tree to achieve superior performance compared to a single decision tree. XGBoost trees typically use a set of manually-extracted features [8], but we have found that supplying the raw CIR to XGBoost trees yields comparable results to an SVM, while not relying on manually-extracted features[3].

SVM implementation. We use the *scikit learn* library[4] with its default parameters, specifically a radial basis function as the kernel and a C value of 1. The employed SVM uses manually-extracted features from 172 samples of the CIR, where the FP index is set to 20[5].

XGBoost implementation. We use the library by *dmlc/xgboost*[6] and use 172 samples of the CIR as input with the FP at index 20[7].

[2]Our choice is driven by the simplicity of these two models. In contrast, deep neural networks are harder to interpret and require longer training times.
[3]We have derived the best hyper-parameters using a grid search approach.
[4]SVM: https://scikit-learn.org/stable/modules/svm.html$#$svm, version 1.0.2.
[5]The features are similar to the ones used in [7, 11], and include the total energy of the CIR, maximum peak amplitude, rise-time, mean excess delay, root mean squared excess delay spread, curtosis, distance between the FP and the mean excesss delay, range, and standard deviation of the noise before the FP.
[6]XGBoost https://xgboost.readthedocs.io/en/stable/, version 1.6.1.
[7]The parameters are eta=0.3, gamma=0.3, #estimators=60, subsample=1, classification objective=reg:squarederror, regression objective=reg:squarederror, and max depth=10.

Dataset	Method	NLOS classification				NLOS error correction			
		F1	Accuracy	Precision	Recall	R^2	MAE LOS	MAE NLOS	Uncorr. MAE LOS/NLOS
BR	SVM	0.89	0.89	0.92	0.86	0.45	0.22	0.23	0.19 / 0.41
	XGBoost	0.86	0.86	0.89	0.83	0.49	0.13	0.22	0.19 / 0.41
ST	SVM	0.82	0.80	0.85	0.85	0.46	0.31	0.44	0.09 / 0.71
	XGBoost	0.77	0.75	0.72	0.83	0.5	0.17	0.41	0.09 / 0.71

Table 1: Baseline performance of SVM and XGBoost. We perform in-dataset analysis using the BR and ST datasets.

3.3 Reused Public Datasets

In our evaluation, we consider four publicly-available datasets.

Dataset BR. Bregar et al. [4] published two datasets recorded in residential and office environments: one for NLOS classification (BR1) and one for NLOS error correction (BR2). The former does not contain true range information and hence no NLOS error correction can be performed. Dataset BR2, instead, can be used for both classification and correction. Both datasets use channel 2, and do not contain info about the type of obstacles between devices.

Dataset ST. Stocker et al. [11] published a dataset suitable for both NLOS classification and error correction. The measurements, collected on channel 5, are labelled as LOS, WLOS (weak LOS) for small obstacles such as monitors, chairs, or people, and NLOS for larger obstacles such as concrete walls and metal doors. Since other datasets do not distinguish between three classes, we merge WLOS and NLOS into a unified NLOS label.

Dataset AN. Angarano et al. [1] published a dataset for NLOS error correction. As discussed in § 4.1, we later omit this dataset, as it does not provide preamble accumulation count information, which is essential to normalize the CIR for comparison to other datasets.

3.4 Experimental Results

We discuss the results of our experimental campaign, starting with a description of the considered performance metrics.

Performance metrics. NLOS error correction results are evaluated using the R^2 *score*[8] and the mean absolute error (MAE). NLOS classification results are evaluated using four common metrics, namely the *F1*, *accuracy*, *precision*, and *recall score*[9]. All classification scores are weighted, so to maintain a balance across LOS/NLOS classes and avoid a biased score in case of unbalanced datasets. Note that scores close to 1 and mean absolute errors close to 0 are desired.

Baseline performance. Before assessing how the considered ML models perform in our testbed, we quantitatively evaluate their NLOS classification and error correction performance when training and testing them on the same dataset, i.e., we perform an *in-dataset* evaluation[10]. Tab. 1 shows the performance of SVM and XGBoost when using the BR and ST datasets[11]. We use these results as a baseline to gauge how well existing techniques perform when tested on the ATA and TTA datasets collected in our testbed.

[8]The R^2 score (coefficient of determination) is 1 if the error prediction is perfect, 0 when it predicts a constant value, and negative when its prediction is arbitrary.
[9]*Precision* is the ratio of the true positives (TP) and the TP plus false positives. *Recall* is the ratio of TP and the TP plus false negatives. *F1* is the harmonic mean between recall and precision. *Accuracy* is the ratio of TP plus true negatives over all predictions.
[10]We do so by using a 5-fold cross-validation, i.e., we split the dataset in five chunks and train the ML model on four of them, while testing on the fifth chunk. This process is repeated five times until all five chunks are used as testset at least once. Note that this procedure is performed for all *in-dataset* evaluations presented in § 3 and § 4.
[11]Despite re-implementing and re-training the models ourselves, the measured performance aligns fairly well with that reported in the original papers.

Dataset	Method	NLOS classification				NLOS error correction		
		F1	Accuracy	Precision	Recall	R^2	MAE LOS (Uncorr: 0.10m)	MAE NLOS (Uncorr: 0.45m)
BR1	SVM	0.27	0.58	0.94	0.16	N/A	N/A	N/A
	XGBoost	0.24	0.56	0.88	0.14	N/A	N/A	N/A
BR2	SVM	0.77	0.73	0.66	0.92	-0.17	0.25	0.37
	XGBoost	0.74	0.72	0.70	0.78	-0.16	0.14	0.30
ST	SVM	0.78	0.77	0.74	0.82	0.06	0.37	0.40
	XGBoost	0.74	0.70	0.65	085	0.03	0.41	0.45
ATA	SVM	0.27	0.58	0.98	0.16	-0.45	0.5	0.46
	XGBoost	0.43	0.58	0.68	0.31	0.05	0.25	0.31
MD1	SVM	0.80	0.78	0.72	0.90	0.12	0.29	0.38
	XGBoost	0.77	0.74	0.69	0.88	0.17	0.28	0.36
MD2	SVM	0.67	0.74	0.91	0.53	0.06	0.40	0.46
	XGBoost	0.63	0.68	0.75	0.54	0.36	0.26	0.29
MD3	SVM	0.70	0.75	0.90	0.57	0.07	0.38	0.46
	XGBoost	0.67	0.70	0.76	0.60	0.39	0.25	0.30

Table 2: Performance of SVM and XGBoost on the TTA dataset when using individual and mixed datasets for training. The best configurations are highlighted with a gray background.

Testbed performance. We train SVM and XGBoost using the ST, BR, and ATA datasets individually, and test them on TTA. The first four rows of Tab. 2 show the obtained results[12]. The classification performance when using BR1 and ATA is very poor, whereas the one measured using BR2 and especially ST is not too far from the baseline values in Tab. 1. The error correction performance becomes low, with close to zero or negative R^2 scores for all individual datasets. Hence, we seek to mix multiple datasets to boost performance.

Merging datasets. We explore three configurations (listed as the last three rows of Tab. 2): (i) MD1, consisting of ST + BR2, (ii) MD2, consisting of ST + ATA, and (iii) MD3, consisting of ST + BR2 + ATA. We distill four main observations from our results. First, combining ST and BR2 slightly improves both the classification and error correction performance compared to that obtained when training using the two individual datasets. Second, adding ATA to any other dataset reduces the classification performance for both SVM and XGBoost, but increases the error correction performance of XGBoost significantly[13]. Third, the measured performance is rather irregular across the board: only MD1 performs best for *both* classification and error correction on the SVM, whereas different dataset combinations achieve the best classification/correction scores with XGBoost. Fourth, among the ML models, XGBoost exhibits the best error correction performance[14]. Moreover, the performance trend of XGBoost seems more predictable when combining datasets.

4 LESSONS LEARNED

While deriving the results in § 3.4, we have learned several insights and lessons, which we enumerate next. These relate to the selection and pre-processing of a dataset (§ 4.1), to the collection and labeling of data (§ 4.2), and to the combination of multiple datasets (§ 4.3).

4.1 Dataset Selection and Pre-processing

In order to train the ML models, the first step consists in selecting one (or some) of the publicly-available datasets, and in pre-processing them for further evaluation. Although these seem like relatively trivial steps, they hide several pitfalls, as discussed next.

[12]We do not consider measurements taken with the wood panel for classification, as they are misclassified ≈ 50% of the time and only have an MAE of 12 cm (see § 4.2).
[13]This implies that adding training data from the same environment (ATA and TTA are collected in the same testbed) may be necessary to achieve a sufficient level of NLOS error correction. We verify this observation in § 4.3 using also the ST and BR2 datasets.
[14]We believe that providing the CIR as input provides better generalization capabilities than using static features, and we will verify this in future work.

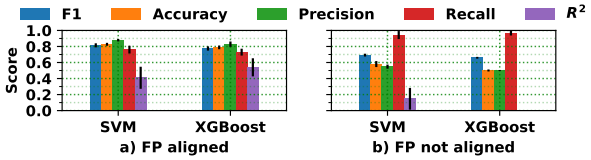

Figure 3: Exemplary impact of FP misalignment on the performance of NLOS classification and error correction. When the FP index of training and testing dataset is not aligned, SVM and XGBoost tend to misclassify LOS as NLOS and the R^2 score drops.

L1: Not all publicly-available datasets are usable. *The lack of preamble accumulation count information in a dataset prevents a correct scaling of the CIR, which causes a significant degradation in the performance of NLOS classification and error correction.*

As outlined in § 2, UWB receivers accumulate several preamble symbols to derive a CIR estimate, and the CIR's absolute values depend on the amount of accumulated symbols. Since the raw CIR as well as features derived from the CIR's amplitude are often used in the training and inference process [1, 4, 7, 11], the performance of several NLOS classification and error correction approaches would highly suffer from a wrong CIR scaling. Hence, if a dataset does not provide preamble accumulation count info, the CIR cannot be scaled, which results in a poor performance. This is the case for the AN dataset, which we hence exclude from our analysis. To exemplify the importance of CIR scaling based on the preamble accumulation count, we perform an *in-dataset* performance analysis using the *ST* dataset by multiplying the CIR values of the testset by a factor of four[15]. Compared to the baseline in Tab. 1, the F1 score of the SVM and XGBoost classifiers drops by 18.3% and 46.9%, respectively.

L2: Not all CIRs are collected equal. *The FP within a CIR is not always at the same index, and can be dissimilar in different datasets. Moving the FP to the same index across datasets is necessary to gain high NLOS classification and error correction performance.*

The DW1000 radio does not always place the detected FP at the same index within the CIR window. As the radio returns the detected FP index, some of the publicly-available datasets pre-process the CIR and move the detected FP to be always at a fixed index (e.g., 5 for [1]). However, the FP index may not be the same across different datasets, and it must be ensured that the FP indexes match when performing cross-dataset evaluation. This is especially important when NLOS algorithms rely on absolute time information within the CIR: an example of this are SVM features such as the mean excess delay of MPCs. Fig. 3(a) shows the performance for the ST dataset when the FP is always aligned at index 20. Fig. 3(b) exemplifies the drop in performance when the FP of CIRs from the testing set are not placed at the same index in every CIR: for these cases the F1 score decreases by up to 16.1%, whereas the R^2 score drops drastically.

L3: The length of the CIR before the detected FP should be sufficiently large. *The samples prior the detected FP contain information that improves the performance of NLOS error correction.*

As outlined in § 2, ranging errors occur when an MPC is misclassified as FP. In some CIR samples taken under NLOS, we observed an increase of the CIR's amplitude values even before the misclassified MPC. We conjecture that an increase of the CIR's amplitudes

[15]We chose a value of 4 as this is the expected difference in CIR amplitude when using a preamble that is four times longer, e.g., PSR=128 vs. PSR=512.

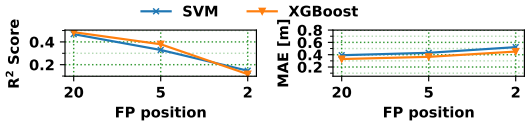

Figure 4: Performance of SVM and XGBoost as a function of the number of CIR samples available before the FP. Considering more samples prior the FP improves the correction performance.

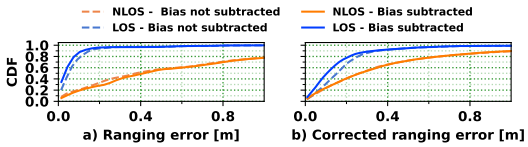

Figure 5: Impact of bias correction. Impact of bias correction on LOS and NLOS ranging measurements prior error correction (a), as well as on the corrected ranging measurements using XGBoost (b).

before the detected FP can be caused either by an attenuated FP or by MPCs, which can be valuable information for NLOS error correction methods. To investigate our assumption, we perform some *in-dataset* analysis on ST, where we test the error correction performance of SVM and XGBoost when either 20, 5, or 2 samples are provided before the detected FP. Fig. 4 shows the results of our analysis: we observe a drop of the R^2 score by 75% and 68% for the XGBoost and SVM models, respectively. This trend is also reflected by an increase of the residual mean absolute error.

4.2 Dataset Collection and Labeling

Also the way data is collected and labelled can largely affect the classification and correction performance, as discussed next.

L4: RSS-dependent distance bias affects the performance of error correction. *The distance estimates of the DW1000 are subject to a distance-dependent error (bias). Correcting this bias on LOS and NLOS measurements increases the correction performance.*

Even in LOS conditions, the DW1000 radio's distance estimate is subject to a received signal strength (RSS) dependent error, called bias, and needs to be corrected. Since the RSS values provided by the DW1000 are not accurate enough, the bias is calculated by using the estimated distance [5]. So far, it has not been studied how the bias correction affects NLOS measurements and whether it improves or worsens the performance of error correction. To answer this, we perform two experiments in which we train and test XGBoost on the ST dataset with and without bias correction. Fig. 5(a) shows the cumulative error distribution of LOS and NLOS measurements prior error correction, revealing that subtracting the calculated bias reduces the error of LOS measurements but increases the error of NLOS measurements. Fig. 5(b) shows the impact of bias correction after using XGBoost to correct the ranging error: the 50 percentile error is reduced by 31.12% in LOS (from 16.2 cm to 11 cm) when the training and testing datasets use bias correction. We conclude that the ML model does not learn how to correct the RSS-dependent bias in the LOS case, but achieves the same NLOS correction with and without bias. The bias correction should hence always be used.

L5: Wrong labeling reduces the performance. *Not all obstacles sufficiently affect the range measurements and manifest in the CIR. Labeling measurements with such obstacles (e.g., wood panels) as NLOS can greatly affect the performance and should be avoided.*

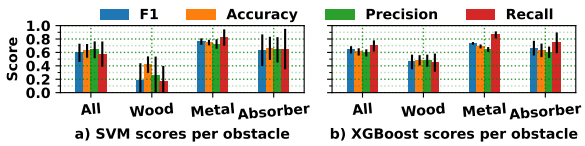

Figure 6: Performance of SVM and XGBoost trained on ST as a function of the type of obstacle. Labeling wooden obstacles as NLOS can significantly decrease the classification performance.

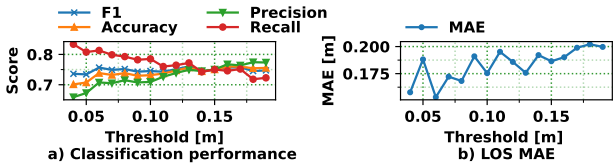

Figure 7: Classification performance and MAE when labeling LOS/NLOS measurements based on the ranging error. When selecting $th_{NLOS} \approx 15$ cm, the classification performance is comparable to that obtained with manual labels (a). A smaller th_{NLOS} during training helps detecting measurements with larger errors (b).

When training a classifier for NLOS classification, it is essential that labeling is correct. Fig. 6 shows the F1, accuracy, precision, and recall scores for SVM (a) and XGBoost (b) when trained on the ST dataset and tested on the combined ATA and TTA datasets as a function of the obstacle type. When evaluating on datasets obtained with all three obstacle types, we can observe that the F1 score is around 0.20 lower compared to the performance obtained performing an *in-dataset* evaluation using ST. Specifically, for both SVM and XGBoost, the classification performance is significantly lower when considering measurements obtained with the wood panel acting as obstacle (its presence is misclassified as LOS in more than 50% of the cases). Indeed, the MAE of NLOS measurements using the wood panel is $\approx$ 12 cm and is close to the typical LOS measurements errors ($\approx$ 10 cm).

L6: Labeling based on the ranging error is a valid alternative to manual NLOS labeling. *We observed on par classification performance when using manually-collected NLOS labels and synthetic labels derived by applying a threshold on the ranging error.*

Automatically generating LOS and NLOS labels based on the ranging error is an effective way to avoid time-consuming and labor-intensive labeling efforts. We show this by labeling a measurement as NLOS whenever the error is larger than a threshold th_{NLOS} and as LOS when it is lower. To find an optimal threshold value th_{NLOS}, we use the ST dataset and sweep th_{NLOS} between 4 cm and 20 cm. For each threshold value, we train and test a model on the whole dataset via 5-fold cross-validation. Fig. 7(a) shows the performance of the XGBoost classifier as a function of th_{NLOS} values, revealing that thresholds around 15 cm are optimal (we have observed a similar trend also with BR2), as the F1 and accuracy scores are at about 75%. Fig. 7(b) shows a positive correlation between th_{NLOS} and the MAE of measurements classified as LOS by XGBoost, i.e., choosing a smaller th_{NLOS} during training makes the ML model more sensitive to detect measurements with larger errors.

L7: Labeling based on the ranging error is not a silver bullet. *Corner cases such as the classification of wooden obstacles as LOS or NLOS remain difficult even when labeling based on the ranging error.*

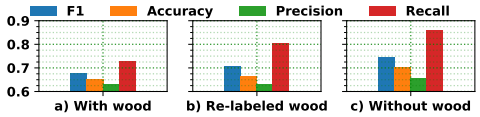

Figure 8: Re-labeling TTA measurements taken with wood panel. Using th_{NLOS}=15 cm (b) helps improving performance compared to the original manual labels (a), but still exhibits a worse performance than omitting the measurements from the dataset (c).

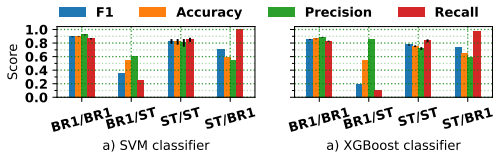

Figure 9: Classification performance of SVM and XGBoost when performing *in-* and *cross-dataset* evaluation. The notation X/Y means training on dataset X and testing on dataset Y.

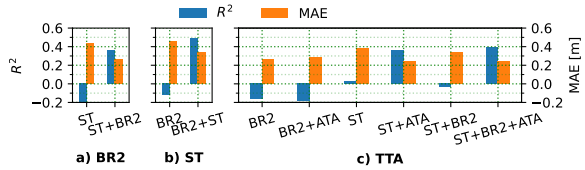

Figure 10: Correction performance when combining datasets with the XGBoost model. Apart from BR2 on TTA, adding training data from a similar environment boosts the correction performance.

We re-label the measurements taken with the wood panel in TTA using th_{NLOS} = 15 cm, and test the performance on the combined ST/TTA dataset. Fig. 8 shows the baseline classification scores with the default wood labels (a), how the F1 score increases by 0.05 when re-labeling the measurements with th_{NLOS}=15 cm (b), but also how the F1 score increases by another 0.05 when filtering out the wood labels completely (c). We hence conclude that re-labeling erroneous labels based on the ranging error can boost performance, but may still be less effective than completely omitting them.

4.3 Evaluations with Crossed & Blended Datasets

§ 3.4 has shown that the performance measured when testing on the TTA dataset is rather irregular across ML methods and various combinations of datasets. We analyze next, whether this is specific to the dataset collected in our testbed, or whether it also applies when performing a cross-dataset evaluation across public traces.

L8: Cross-dataset performance is highly irregular. *Also when training and testing across public datasets taken in different settings, the performance is worse than when performing in-dataset testing.*

Fig. 9 shows the *in-dataset* and *cross-dataset* classification performance of the SVM (a) and XGBoost classifier (b) for different combinations of ST and BR. Both classifiers perform well during *in-dataset* evaluation: the F1 score is 89.1% and 85.5% for SVM and XGBoost, respectively, with the BR1 dataset. Similar results are obtained when using ST, with an F1 score of 82.4% and 77.1% for SVM and XGBoost. The performance is vastly different when performing *cross-dataset* evaluation. When training on BR1 and testing on ST, the F1 score drops to 35.7% for SVM and 19% for XGBoost. The drop is not symmetric: when using ST for training and BR1 for testing, the F1 score is around 70–75% for both models. When evaluating the error correction performance, we observed that when training on ST and testing on BR2, the MAE of LOS/NLOS measurements is 19/54 cm, compared to 17/41 cm when performing an *in-dataset* evaluation. The performance is worse when training on BR2 and testing on ST: the MAE of LOS/NLOS measurements is 38/48 cm, compared to 13/22 cm when performing an *in-dataset* evaluation.

L9: Adding traces collected in similar environments for training is highly recommended when performing correction. *Adding training data from the testing environment may be needed to improve the performance of error correction to an acceptable level.*

Fig. 10 shows that adding training data from BR2 to ST improves the cross-dataset correction performance of XGBoost when testing on BR2 in terms of an increased R^2-score and reduced MAE (a). The same trend is visible when adding training data from ST to BR2 while testing on ST (b), as well as when using mixed datasets and testing on TTA (c). This trend is not visible when using SVM.

5 CONCLUSIONS AND FUTURE WORK

We evaluate the NLOS classification and error correction performance obtained with SVMs and XGBoost trees using public datasets and traces collected in our UWB testbed. After presenting our results, we summarize key insights we have learned during this study, hoping they will be useful to researchers and practitioners approaching this research area for the first time. As next steps, we plan to add DNNs to the set of benchmarked ML models, and to analyze how to best perform automatic labeling based on the ranging error.

ACKNOWLEDGMENTS

This work was supported by the TU Graz LEAD project ("Dependable IoT in Adverse Environments") and by the ENHANCE-UWB project ("Benchmarking and Advancing Localization and Communication Performance of UWB Systems in Harsh Environments"). This work was also partially executed within the SPiDR project ("Secure, Performant, Dependable, and Resilient Wireless Mesh Networks") financed by the Technology Innovation Institute.

REFERENCES

[1] S. Angarano et al. 2021. Robust UWB Range Error Mitigation with Deep Learning at the Edge. *Engineering Applications of Artificial Intelligence* 102 (2021).
[2] V. Barral et al. 2019. NLOS Identification and Mitigation Using Low-Cost UWB Devices. *Sensors* 19, 16 (2019).
[3] J. Borras et al. 1998. Decision Theoretic Framework for NLOS Identification. In *Proc. of the 48th VTC Conf.*, Vol. 2.
[4] K. Bregar et al. 2018. Improving Indoor Localization Using Convolutional Neural Networks on Computationally Restricted Devices. *IEEE Access* 6 (2018).
[5] Decawave. 2017. DW1000 User Manual, version 2.11.
[6] İ. Güvenç et al. 2007. NLOS Identification and Weighted Least-Squares Localization for UWB Systems Using Multipath Channel Statistics. *JASP* (2007).
[7] S. Maranò et al. 2010. NLOS Identification and Mitigation for Localization Based on UWB Experimental Data. *IEEE JSAC* 28, 7 (2010).
[8] A. Musa et al. 2019. A Decision Tree-based NLOS Detection Method for the UWB Indoor Location Tracking Accuracy Improvement. *Int. J. Comm. Syst.* 32 (2019).
[9] J. Schroeder et al. 2007. NLOS detection algorithms for Ultra-Wideband localization. In *Proc. of the 4th WPNC Worksh.*
[10] M. Schuh et al. 2022. First Steps in Benchmarking the Performance of Heterogeneous Ultra-Wideband Platforms. In *Proc. of the 5th CPS-IoTBench Worksh.*
[11] M. Stocker et al. 2021. Performance of SVR in Correcting UWB Ranging Measurements under LOS/NLOS Conditions. In *Proc. of the 4th CPS-IoTBench Worskh.*
[12] V. Tran et al. 2022. DeepCIR: Insights into CIR-based Data-driven UWB Error Mitigation. In *Proc. of the IROS Conf.*
[13] H. Wymeersch et al. 2012. A Machine Learning Approach to Ranging Error Mitigation for UWB Localization. *IEEE Trans. on Communications* 60, 6 (2012).
[14] J. Xin et al. 2019. A Bayesian Filtering Approach for Error Mitigation in Ultra-Wideband Ranging. *Sensors* 19, 3 (2019).