IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 January 2024, accepted 22 February 2024, date of publication 29 February 2024, date of current version 7 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3371907

RESEARCH ARTICLE

Novel Classification Method to Predict the Accuracy of UWB Ranging Estimates

MERITXELL ARSUAGA[1], AITOR OCHOA-DE-ERIBE-LANDABEREA[2,3], LETICIA ZAMORA-CADENAS[2,3], SAIOA ARRIZABALAGA[2,3], AND IGONE VELEZ[2,3]

[1]Ibermática an Ayesa Company, 20009 Donostia/San Sebastián, Spain
[2]CEIT-Basque Research and Technology Alliance (BRTA), 20018 Donostia/San Sebastián, Spain
[3]Tecnun School of Engineering, Universidad de Navarra, 20018 Donostia/San Sebastián, Spain

Corresponding author: Aitor Ochoa-de-Eribe-Landaberea (aochoadeeribe@ceit.es)

This work was supported in part by Ministerio de Ciencia, Innovación y Universidades of the Spanish Government, through the Project PICRAH 4.0, under Grant PLEC2023-01353; and in part by the Basque Government through the Project B-Ind5G under Grant KK-2021/00026 and through the Project u4Smart under Grant KK-2023/00016.

ABSTRACT Real time location systems (RTLSs) are becoming more relevant in a more data driven economy and society due to their wide range of application cases. When the location of an object needs to be tracked with high accuracy, ultra wideband (UWB) technology is usually the best option. Nevertheless, UWB ranging estimates are not completely immune to some sources of error such as non line of sight (NLOS) or multipath conditions. Thus, this paper proposes a real-time classification model based on machine learning (ML) to predict if received ranging estimates are in line of sight (LOS) or NLOS conditions and discard those in NLOS. However, it is also shown that classifying measurements as LOS or NLOS does not guarantee detecting inaccurate ranging estimates, since LOS measurements can also yield large errors. As an example, the ranging root mean square error (RMSE) of the data labelled as LOS in a UWB based localization system database in the literature is of 0.714 m, significantly higher than the theoretical accuracy of a UWB system. Thus, a novel ML-based classification model is proposed to predict the magnitude of the ranging error. After applying the proposed classification model in the same data, the ranging RMSE of those ranging samples classified as most accurate is of only 0.183 m, significantly lower than the best RMSE we can obtain on the classical LOS/NLOS classification approach.

INDEX TERMS DWM1000, machine learning, random forest, ranging errors, real time location system, ultra wideband.

I. INTRODUCTION

Industry 4.0 is revolutionizing the way companies manufacture and deliver their products. Among all the recently emerging technologies, real-time location systems (RTLSs) are gaining relevance in industrial and home use. RTLSs can enable the autonomous operation of vehicles and robots to improve the productivity and flexibility of production networks [1], [2], [3] and measure the trajectories of assets for data analytics [4]. Apart from the productivity advantages, RTLSs can also be used to enhance the safety of people at work [5], [6] or at home with ambient assisted living (AAL) applications [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Moinul Hossain.

There are many technologies which can be used for the development of an RTLS. Most of them are based on radio-frequency technology since it makes the identification and tracking of objects and people an easy task [6]. The most common radio-frequency-based technology is global navigation satellite system (GNSS). However, it lacks the necessary accuracy in many applications, especially in indoor environments [8]. For real-time tracking in indoor environments, other radio-frequency-based RTLSs such as bluetooth low energy (BLE), ultra wideband (UWB) or Wi-Fi have better accuracy [9].

An RTLS based on radio-frequency technology contains two main elements: anchors and tags. Anchors are fixed sensors at known locations, whereas tags are moving sensors with unknown positions. The positions of tags are calculated by means of ranging estimates between each anchor and tag. Those ranging estimates can be obtained using time, angle or power information. The problem of radio-frequency-based RTLSs is their sensitivity to non line of sight (NLOS) conditions and multipath effects, usually caused by congested environments with metallic objects. This is especially true in factories. In fact, huge errors in ranging estimates may occur under NLOS and multipath conditions, with negative consequences in the localization accuracy. Among the mentioned technologies, UWB presents more robustness against NLOS and multipath conditions [10]. However, UWB can still have bad ranging estimates, which need to be detected and discarded or corrected.

There are several methods to detect bad ranging estimates. For example, previous knowledge of moving constraints can be used as in [11]. By calculating the variance of the difference between consecutive ranging estimates in a sliding window and comparing it with a threshold, bad ranging estimates can be detected. However, the threshold with which to compare the variance is dependent on the maximum velocity of the tracked object, which must be known. Moreover, the known maximum velocity must be low. That is the reason why the proposal of [11] is thought for indoor robots moving at 1 m/s. In other scenarios, it cannot be known if a sudden change in a ranging estimate is the result of a bad measurement or the consequence of a high moving velocity.

Other possibility is to combine inertial measurement unit (IMU) data with UWB ranging estimates and detect bad ranging estimates using the Mahalanobis distance as in [12]. However, this approach needs to add extra sensors and have an excellent model of the noise of these sensors.

In recent years, machine learning (ML) techniques have gained popularity in order to improve the accuracy of UWB-based RTLSs. The advantage of ML techniques is that bad ranging estimates can be detected on time only with the information extracted from the received signal. No extra sensors or previous knowledge of moving constraints are necessary. In order to detect bad ranging estimates, References [13], [14], [15], [16], [17], and [18] propose ML models that predict if received signals are in line of sight (LOS) or NLOS conditions. Others propose to further separate NLOS measurements making multiclass classifications [19], [20], [21]. However, the main focus of the mentioned proposals [13], [14], [15], [16], [17], [18], [19], [20], [21] is to improve the classification performance of UWB ranging estimates. None of them focuses on their real-time applicability. If the proposed models are to be applied in an actual RTLS, the classification time per sample should be lower than the time between two consecutive measurements.

Knowing the importance of real-time applicability, other proposals measure the time spent per sample [22], [23], proving that ML classification models can be used in an RTLS. These classification models separate LOS and NLOS [23] and LOS, NLOS and multipath conditions [22]. The problem of LOS/NLOS classifications [13], [14], [15], [16], [17], [18], [23] or their derivations [19], [20], [21], [22] is that not all NLOS measurements give bad ranging estimates. The severity of NLOS conditions are not equal in different environments, and some of them might actually give acceptable accuracy. Thus, if such a classification model is used in an RTLS, some accurate ranging estimates can be discarded, which can imply a reduction in accuracy. Moreover, some LOS measurements can be incorrectly labelled if an unexpected reflection distorts the measurement. This also would imply a reduction of accuracy.

A better approach than the LOS/NLOS classification is to make a regression to predict the ranging error, as in the case of [21]. However, their regression model is highly dependent on the LOS, Hard-NLOS or Soft-NLOS condition. Thus, for a correct application of a regression, a previous classification is needed, making necessary the use of additional calculations, increasing the computing time and reducing the real time applicability.

As an alternative to the state of art, the contributions of this paper are twofold. First, if LOS/NLOS classification were necessary with UWB measurements, we propose a novel classification model that works in real-time and performs better than [23]. We choose [23] as reference because it is the only work proposing an LOS/NLOS classification model while taking into account the computational burden. Although our proposal obtains better performance, one of the findings of this manuscript is that accurately detecting NLOS does not guarantee detecting all bad ranging estimates. Thus, the main contribution of this paper is a novel classification model that separates ranging estimates as Good, Medium or Bad, which is a more suitable classification for an RTLS. Up to the authors' knowledge, this is the first time that such a classification model is proposed. Moreover, we prove that the proposed model makes a good trade-off between performance and real-time applicability. The results show that the processing time is so short that the proposed classification model can be applied in real-time.

The rest of the article is organized as follows. Section II shows how our proposed ML classification models can be used in an RTLS and describes the proposed models. Section III describes the followed methodology, including the used data and evaluation criteria. Finally, Section IV shows and discusses the obtained results and Section V gives the final conclusions.

II. RTLS WITH PROPOSED CLASSIFICATION MODELS

The proposed ML-based classification models are intended to be used with DWM1000 modules of Qorvo-Decawave. These modules can be configured to calculate ranging estimates among them with the two way ranging (TWR) algorithm [24]. This algorithm consists of measuring the propagation time between two transceivers configured as a tag and an anchor. With the measured propagation time and the known speed of light, the distance between each anchor and tag can be estimated.

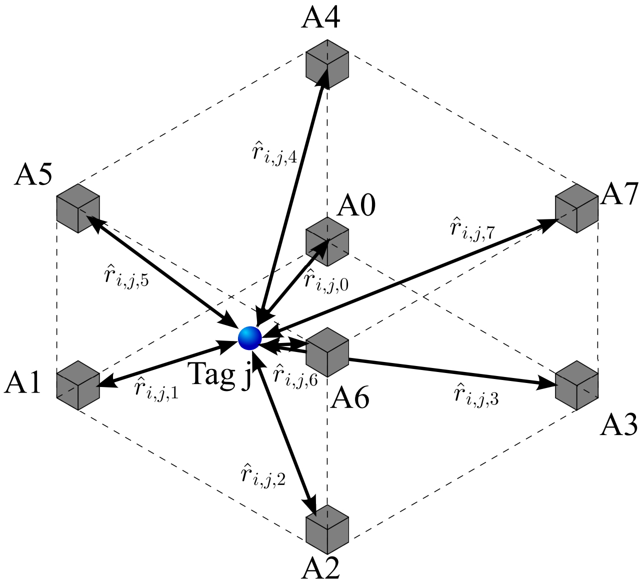

FIGURE 1. Basic architecture of radio-frequency-based RTLS using time of flight (ToF) measurements.

With the configured UWB transceivers, an RTLS such as the one shown in Figure 1 can be set up. In the figure, there can be seen some anchors represented as grey cubes and a tag as a blue sphere. During the operation, anchors and tags run the TWR algorithm among them to compute all the ranging estimates. The ranging estimates are represented as $\hat{r}_{i,j,n}$ for the calculated distance at time instant i between tag j and anchor n. Since the positions of anchors are previously known, the position of tag j at time i, $\hat{p}_{i,j}$, can be computed using the anchors' positions along with the ranging estimates between all anchors and tag j at time instant i.

At least four ranging estimates are needed at the same time to calculate the three-dimensional position of a tag. This position is calculated as the intersecting point among four known spheres. Ideally, the position of the tag can be algebraically calculated by solving the corresponding equation system [25]. However, in the real world, measurements are noisy and statistical approaches such as the least-square method or extended Kalman filter are preferred [26].

The least-square method and extended Kalman filter work well when sensors have Gaussian noise [27], [28]. This is usually the case with UWB ranging estimates under LOS conditions. Nevertheless, multipath and NLOS conditions can produce unexpected errors that do not follow a Gaussian distribution. If these bad ranging estimates enter in the positioning algorithm, the provided position can have a considerable error. In order to obtain an accurate positioning performance, early detection of bad ranging estimates is crucial so that they do not affect the final position estimate.

In order to improve the positioning accuracy, this paper proposes an ML-based classification algorithm to detect and discard bad ranging estimates on time. Figure 2 shows the flow chart of an RTLS with our proposed classification model. The RTLS first uses the TWR algorithm to calculate

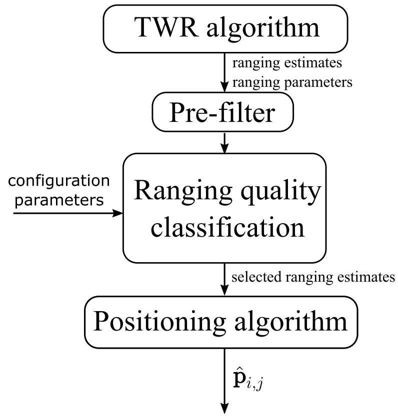

FIGURE 2. Flow chart of an RTLS that classifies ranging estimates according to the magnitude of the distance error and discards the worst of them.

TABLE 1. UWB parameters used in classification models.

Ranging parameters	
RANGE	Ranging estimates between anchor and tag
FP_INDEX	First path index
RSS	Received signal strength of the frame
RSS_FP	Received signal strength of the first peak
FP_POINT1	First path amplitude point 1
FP_POINT2	First path amplitude point 2
FP_POINT3	First path amplitude point 3
STDEV_NOISE	Standard deviation of noise
CIR_POWER	Channel impulse response power
MAX_NOISE	Maximum reported noise
RXPACC	Preamble accumulation count
Configuration parameters	
CHANNEL_NUMBER	Number of the UWB channel used for communication
FRAME_LENGTH	Frame length
PREAMBLE_LENGTH	Preamble length
BITRATE	Bit rate for the data portion of the frame
PRFR	Pulse repetition frequency
PREAMBLE_CODE	Preamble code chosen according the UWB channel and pulse repetition frequency

the ranging estimates between each tag and anchor as well as other parameters related to the received signal characteristics. All these parameters are shown in Table 1 and are denoted as ranging parameters. The ranging algorithm can sometimes have some errors that produce inconsistent data such as negative ranging estimates. Thus, all negative ranging estimates are discarded with a pre-filter as proposed by [2]. Then, those ranging estimates pass through our proposed classification model to be classified and let the positioning algorithm decide to include them or not. The classification model uses the ranging parameters defined in Table 1 as well as some configuration parameters related to the transmitted signal characteristics such as the channel number. The configuration parameters that the classification model uses are also shown in Table 1. Finally, the positioning algorithm calculates the position of tag j at time i, $\hat{p}_{i,j}$, using the best ranging estimates.

A. PROPOSED ML CLASSIFICATION MODELS

The main contribution of this paper is an ML-based classification model to detect bad ranging estimates in an

RTLS. This classification model classifies received UWB ranging estimates as Good, Medium or Bad according to the ranging error magnitude. Usually, DW1000 transceivers obtain ranging errors under 15 cm in clear LOS conditions, as in the case of [29], [30]. Thus, measurements with an error of less than 15 cm are categorized as Good in this research work. Nevertheless, under more challenging conditions it might be difficult to obtain enough ranging estimates of good quality. For this reason, if necessary, we will accept those ranging estimates with an error up to twice the Good class threshold, i.e. 30 cm. In fact, it might also be possible to obtain errors of this magnitude under optimal conditions, as [30] obtained a maximum error of 28.5 cm. Thus, we will label ranging estimates with an error between 15 and 30 cm as Medium class. Finally, measurements with a greater error, exceeding 30 cm, are categorized as Bad and shall be discarded. Note that we could have defined a classification model of more levels or even made a regression model. Nevertheless, a trade-off between model performance and real-time applicability is needed.

If it were necessary to classify UWB measurements as LOS or NLOS, we also propose an ML-model that predicts if measurements are LOS or NLOS. Although each of the models are trained to classify different class types, both of them are generated following the same process described in the following subsections.

1) USED PARAMETERS

For the creation of ML models, the parameters of Table 1 are used. Most of these parameters can be easily obtained with DWM1000 transceivers.

2) CLASSIFICATION MODEL

The proposed ML models use a Random Forest classification introduced by Breiman in [31]. The concept of these models is based on an ensemble in which multiple classifiers are combined in order to solve a more complex problem and improve model performance. The classifiers that create this ensemble are decision tree type. Each of this is trained with a slightly different sample of the training data. The results obtained from these individual trees will determine the final prediction.

3) HYPERPARAMETER TUNING

The cross-validation technique is used for model tuning and selection. This method is used for evaluating ML models, which is based on dividing the training data set into k subsets. In this way the accuracy of the model is measured in a more realistic way.

Each ML model has a set of hyperparameters [32] that serve to fit the model to the data. Both the model built and the results obtained will depend to some extent on the values assigned to these hyperparameters. These values should be set before training and will depend on the profile of the data being analyzed. There are different techniques for the selection of these values. The technique used in this case is the grid search, as it is a very exhaustive and easy to implement method. This method is based on the search for the optimal values of the hyperparameters by analyzing all possible combinations of these. The selected values will be those with which the best results are obtained. First, a broad and exhaustive range of parameters is selected. For each combination of these values, a Random Forest classification model is trained and evaluated using the cross-validation technique. It is important to assign a metric on which to base the choice of the combination of hyperparameters that gives the most accurate results. As it can be deducted from the description of grid search, this technique needs many computations, especially if the search space is large. As random forest models usually work reasonably well with default settings [32], the search space is small and the grid search becomes a good option for hyperparameter tuning in our case.

The Random Forest model has the following hyperparameters to be tuned:

- *num trees*: refers to the number of decision trees that will build the final predictive model.
- *mtry*: the number of predictors to sample at each split.
- *min node size*: minumun number of instances in terminal node.
- *splitrule*: The rule by which each split is considered in a tree, the impurity measure to separate to one class from another in the target variable.

The *num trees* hyperparameter value selection will be based on testing with different numbers of trees within the search space, selecting the value with which the best results are obtained. Once the number of trees has been selected, a search is carried out by testing different combinations between the other three hyperparameters. The combination of values to be selected will be the one with which the best results are achieved.

4) FEATURE SELECTION

The success of a model resides in the quality of the data from which it will learn. If irrelevant and noisy features compose the data, ML algorithms could predict results less accurate and difficult to understand. Feature selection is the process of identifying a subset of features to be used in model building. These techniques allow simplifying the models in order to make them more interpretable, decrease training time and help to reduce model overfitting.

The technique used is called Recursive Feature Elimination (RFE) [33], an efficient algorithm whose objective is the selection of a subset composed of the most relevant attributes for the construction of the predictive model being worked on. RFE works iteratively by removing the least significant parameter in each step. The importance of each feature is gauged by assessing the performance drop of the model when compared between the full set of features and a model excluding a particular feature. In a model encompassing p features, RFE generates p submodels, yielding p importance

TABLE 2. Constant features.

Feature	Value
FRAME LENGTH	39
PREAMBLE LENGTH	4096
BITRATE	110
PRFR	64

evaluations. After removing the least important feature, it is important to reassess the new set of features, as the feature importance can change substantially in each iteration.

The reason for choosing the RFE method is its good performance with an acceptable computational burden compared to other techniques such as Filter Methods, regularization techniques and Best Subset Selection [34], [35]. Filter Methods require very little computational power, but usually have a poor performance. On the other side of the spectrum is the Best Subset Selection, which is a brute force method that tests all possible subsets from a given set of features. Regularization techniques are less burdensome than the Best Subset Selection, but add a new hyperparameter to tune.

III. METHODOLOGY

A. USED DATA

For the training of reliable ML models, extensive measurements must be made. The work of [23] published a large database with high variety of conditions. As the measurements were taken with DWM1000 transceivers, we use their published database to train our proposed ML-based classification models. This will also allow a fairer comparison with the algorithm proposed by [23].

The database of [23] was obtained with UWB measurements between a tag and eight anchors. The UWB localization system was tested in four different environments and in six different UWB channels to avoid data over-fitting. The tested environments consisted of an apartment, a house, an industrial facility and an office, each one containing specific multipath propagation characteristics. This abundance of measurement conditions helps the proposed models to be effectively used under different conditions such as a new environment. In all these experiments, the eight UWB anchors were placed at known fixed locations, and the tag was placed at many different locations, resembling a human walking path. All in all, 491 040 data samples were obtained. These data contain 26 variables and the parameters of Table 1 are among them.

1) DATA CURATION

In order to start working with the data and apply ML algorithms to them, the first step is to make sure that the data are in the right form. A pre-process is performed for this, which consists of checking the existence of null or missing values, constant columns or any outlier. It can be seen that there are four columns with constant values, shown in Table 2.

On the other hand, the variables TAG_ID, ANCHOR_ID, X_TAG, Y_TAG, Z_TAG, X_ANCHOR, Y_ANCHOR and Z_ANCHOR are removed. Some of these features have been used in the error calculation by obtaining the real ranging value. The rest are anchor and tag identification numbers that do not provide information to get the predictions.

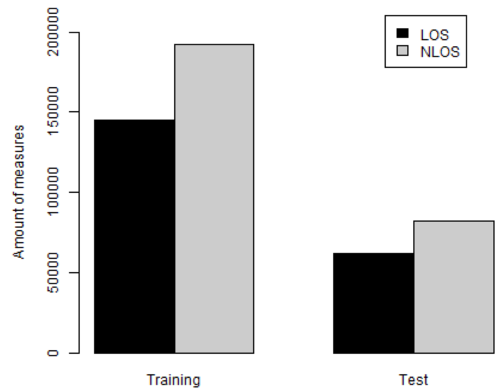

FIGURE 3. NLOS and LOS in training and test set.

It is also noted that some measurements have negative values of the RANGE column, which would have been automatically removed by the pre-filter mentioned in Section II. Thus, we filter out these values, leaving the data set with only positive RANGE values.

After applying the above mentioned pre-filter, the final data set to work with contains 482 361 rows and the 13 features described in Table 1.

2) DATA SUBSETS

Once the database is in the right form, it is used for the training and validation of ML models. For the validation of the ML models, it is necessary to have a subset of the data set that has not been used creating the model, i.e., this subset will be composed of observations from which the model has not learned. For this purpose, the data set is divided into a training subset (70% of the data) and a test subset (30% of the data).

For the model to learn based on all types of measurements, it is important that the training and test subsets contain balanced classes. In the case of LOS/NLOS classification, Figure 3 shows the number of observations corresponding to each class in training and test subsets. All in all, 43% of observations are LOS and 57% NLOS.

Another important characteristic of the training and test subsets is that the distribution of the environments is as homogeneous as possible. Figure 4 shows how a similar distribution has been maintained in both subsets.

For the generation of the Good/Medium/Bad classifier, another separation of the data set is made to maintain similar proportions of the target feature in each of the subsets. Figure 5 shows how in the present case the data set is imbalanced. The Bad class is the class for which more information is available, so the model will detect bad ranging estimates better than others. This is not a problem since the objective of the proposed ML model is to reliably detect and discard bad ranging estimates.

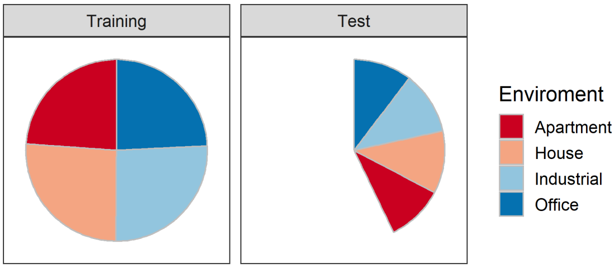

FIGURE 4. Environments in training and test sets.

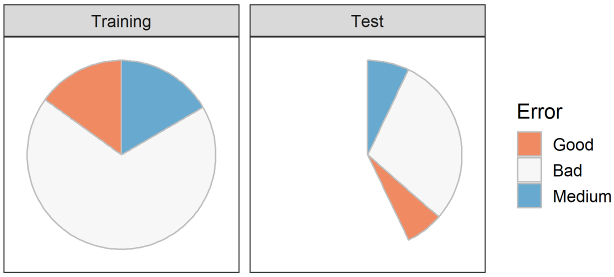

FIGURE 5. Error classes training and test sets.

TABLE 3. Confusion matrix.

Prediction	Positive	Negative
Positive	True positive (TP)	False positive (FP)
Negative	False negative (FN)	True negative (TN)

B. PERFORMANCE EVALUATION

In order to carry out the feature selection and the hyperparameter tuning, it is necessary to define the metric to be used. This metric will indicate the quality of the results and will be the base for the decisions to be made.

The data set is not balanced in any of the models, since the proportions of the different classes are not the same. For this reason, the selected metric is the F1-score. This metric takes values between zero and one and combines the Precision or Positive Predictive Value (PPV) and Recall or True Positive Rate (TPR) metrics by means of the harmonic mean. The Precision measures the quality of the model and the Recall metric reports the amount that the Machine Learning model is able to identify correctly. These values can be calculated by means of the confusion matrix, shown in Table 3.

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \tag{1}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \tag{2}$$

$$\mathrm{F1} = \frac{2\cdot\mathrm{PPV}\cdot\mathrm{TPR}}{\mathrm{PPV}+\mathrm{TPR}} \tag{3}$$

Unlike the LOS/NLOS classification model, the proposed classification model based on the distance error of ranging estimates deals with a multiclass classification. In this case, the confusion matrix needs a slight modification to represent all three classes: Good, Medium and Bad. This confusion matrix contains 3×3 elements to show how each sample belonging to these classes have been classified. In order to obtain the classification performance metrics from a 3×3 confusion matrix, the evaluation must be performed in three steps. First, the Good class is considered the positive case and Medium and Bad classes are negatives. Making this assumption, precision, recall and F1-score are calculated for the Good class. Second, the Medium class is considered positive and Good and Bad negatives to calculate the performance metrics for the Medium class. Third, the same is done with the Bad class. Once the values of Precision, Recall and F1 are calculated for each of the classes, the average of these is obtained.

Although the F1-score is a suitable parameter to evaluate a classifier, we must not forget that the proposed classification model is intended to select the best ranging estimates. Thus, we will evaluate the proposed model with ranging error metrics obtained for each predicted class. Given a set of ranging samples classified in a certain class, its average ranging error, the standard deviation of the ranging error, the ranging root mean square error (RMSE) [36], the maximum ranging error in 95% of cases and the absolute maximum ranging error are going to be calculated.

IV. RESULTS

A. LOS/NLOS CLASSIFICATION

Once the methodology described above has been defined, it has been applied to the data set. Firstly, the ability to predict whether a measure is of NLOS or LOS type is studied using the total of the input variables of the data set. Subsequently, these results have been improved by using a reduced set of variables, obtaining a more flexible and simpler model.

1) HYPERPARAMETER TUNING

The first hyperparameter to be set for the generation of the classification model consists of the number of decision trees that will constitute the Random Forest model. The cross-validation technique is used for model tuning and selection. In this case, the training data set is divided into 10 folds.

Figure 6 shows the values of the F1 metric measured for different values of the *num trees* hyperparameter. When using cross-validation, there will be 10 score values, one for each iteration. The values shown in the graph are the average of the 10 F1-scores calculated with each fold. On the one hand, the F1-score obtained with the training data set with which the model is created is observed. On the other hand, the F1-score obtained with the validation set of each iteration has been evaluated. Figure 6 shows that with the construction of a few trees the F1 value tends to converge. Although it also shows that adding more trees is not detrimental to the results, the computation time increases considerably when adding more trees. For these reasons it is decided to set the number of trees to 200.

Once the number of trees to be used has been fixed, the next step is the grid search for the rest of the hyperparameters. This way of selecting the optimal values of the hyperparameters allows performing a sensitivity analysis of them. As seen with

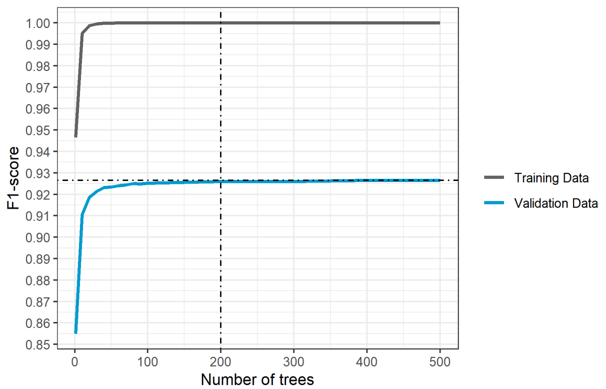

FIGURE 6. ***num trees*** **selection in NLOS/LOS classification.**

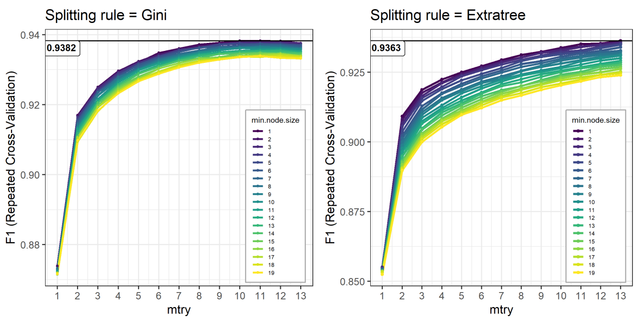

FIGURE 7. **Hyperparameter tuning in NLOS/LOS classification.**

the hyperparameter *num trees*, Figure 7 shows that the model is also strongly influenced by other hyperparameter values. Nevertheless, it can also be seen that not all of them affect in the same way the results, being the model more sensitive to some hyperparameters than to others.

First, the influence of the splitting rule used during tree construction is evaluated. The subgraph on the left of Figure 7 shows obtained results when using the Gini rule [37], with which the best results are obtained. The right subgraph shows the results for the case in which the Extratree splitting rule is used in the construction of the trees. This rule shows two main differences compared to other methods based on sets of trees, which are that it divides the nodes by choosing the cut points completely randomly and that it uses the entire learning sample.

The Gini splitting rule consists of selecting among all the splits of the candidate variables (*mtry*), the one that minimizes the Gini impurity. As shown, the greater the number of variables to be taken into account in each division, better results will be obtained by the model. Finally, it is observed that the model is not as sensitive to the hyper parameter that will determine the complexity of the decision trees (*min node size*) as it is to *mtry* and *num trees*. Despite the difference being smaller, it is observed that the best results are obtained with low values.

Table 4 shows the values selected for each hyperparameter, with which the predictive classification model will be built. By means of these hyperparameters the model obtains an F1 of 0.9382 through cross-validation. On the other hand, the time needed to build the complete model will be 263.19 seconds.

TABLE 4. **Selected hyperparameters in NLOS/LOS classification.**

	num trees	mtry	min node size	splitrule
Tuned value	200	10	1	Gini

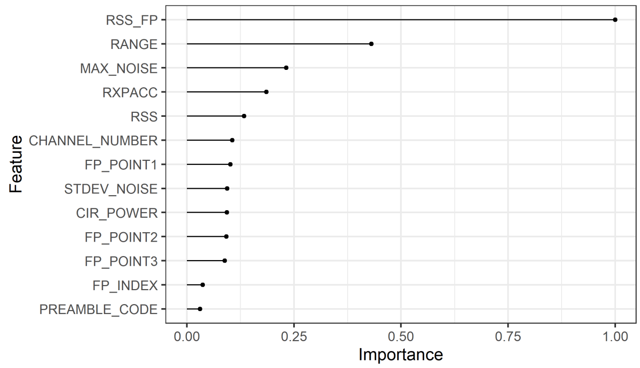

FIGURE 8. **Feature importance in NLOS/LOS classification.**

2) FEATURE SELECTION

In order to interpret the predictive model more accurately, the importance of each input feature is studied to make the predictions. The purpose of this analysis is to eliminate the most irrelevant features, thus improving the accuracy of the model. Figure 8 shows the importance of each feature.

The feature selection process was carried out by means of this analysis. As mentioned above, the technique used was RFE, which reduced the number of input features of the classification model. Figure 9 shows the F1 obtained by repeated cross-validation in the model created in each iteration. In all these models, the number of input features shown on the x-axis is used. It is observed that when 7 or more features are used, the accuracy of the model remains at similar values. With this set of features the F1 obtained in the cross-validation phase is 0.9399982. Using these 7 variables, the other 6 that are eliminated do not contribute any additional value to the model, since the accuracy does not improve significantly.

This new reduced subset consists of the features shown in Figure 10. It also shows the importance of each of these features in the new classification model created. As mentioned above, the RFE can substantially change the importance of some features, as is the case in Figure 8 and Figure 10.

3) MODEL PERFORMANCE

The final results obtained with both the total features (TF) and the selected features (FS) are shown in Table 5, where the positive class refers to NLOS type measures. The two models were created with the same training data set. Subsequently both have been tested with the test data set, composed of 144 708 measurements. All the work has been run on a

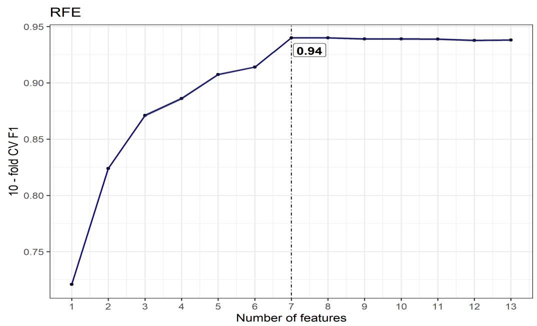

FIGURE 9. RFE Feature Selection in NLOS/LOS classification.

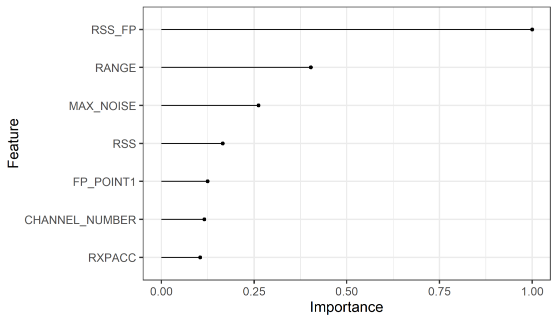

FIGURE 10. Selected subset feature importance for NLOS/LOS classification.

PC with an Intel(R) Core(TM) i5-7500 CPU @3.40GHz processor using 8GB of RAM.

The results obtained are very similar, improving a bit the accuracy of the model composed by the reduced set of features. Although the prediction times obtained are low and it does not seem that there is a great improvement, this can become an essential aspect if a more complex model or larger quantities of measurements to predict were available. The reduced model is able to predict 58 823 measurements per second.

4) COMPARISON WITH LITERATURE

Once the results obtained from the Random Forest models have been analyzed, they have been compared with those presented by [23]. The results of [23] are obtained using raw channel impulse response (CIR) data from the measurements and the best fitting model to these data is a convolutional neural network (CNN). Table 6 shows the metrics obtained with our proposal compared to [23].

As it can be observed, our classification model performs better in Accuracy, Precision, Recall and F1 score. The sample prediction time is a bit more difficult to compare, since we did not use the same computation platform as [23]. Their proposal was tested in various platforms, being the most similar to our set-up the Intel i7-2670QM CPU with 8 threads and 8 GB of RAM. With that set-up, they needed 28.6 μs per sample. In their best case, with an i7-6700HQ CPU with 8 threads and 16 GB of RAM, they needed 19.8 μs per sample. However, sending raw CIR data can be time demanding. Thus, if their classification model was applied

TABLE 5. NLOS/LOS classification performance.

Metric	TF	FS
True Positive	76 995	77 230
False Positive	3059	3201
True Negative	59 183	59 041
False Negative	5471	5236
Accuracy	94.1%	94.2%
Precision	93.9%	93.9%
Recall	94.2%	94.3%
F1	94.0%	94.1%
Sample Prediction time	17.5 μs	17.0 μs

TABLE 6. NLOS/LOS classification performance compared with literature.

Metric	FS (RF)	[23]
Accuracy	94.2%	87.4%
Precision	93.9%	85.9%
Recall	94.3%	89.4%
F1	94.1%	87.6%
Sample Prediction time	17.0 μs	28.6–19.8 μs

in real time, the time needed to send raw CIR data to the central processing unit should also be considered. Even without taking into account this setback of [23], our proposed classification model runs faster.

B. CLASSIFICATION PROPOSAL

Sometimes, accurately detecting NLOS situations does not guarantee detecting bad ranging estimates. In fact, this is the case with the used database as it can be observed in Figure 11, where the cumulative distribution functions (CDFs) of ranging errors are plotted. Blue squares represent data samples of measurements labelled as LOS, while orange triangles show the samples labelled as NLOS. Thus, it can be seen which the ranging accuracy of each group would be if a perfect classification accuracy was obtained. It is noted that the ranging accuracy of ranging estimates are not very different under LOS or NLOS.

For further analysis, some statistics of these errors are shown in Table 7. The columns, from left to right show the average ranging error, the standard deviation of the ranging error, the ranging RMSE, the maximum ranging error in 95% of cases and the absolute maximum ranging error. These statistics confirm what has been observed in Figure 11: there is no significant difference in the obtained errors between the LOS and NLOS measures. Thus, we can conclude that labelling UWB measurement samples as LOS or NLOS presents some problems. NLOS conditions can produce low or large distance errors depending on the severity. Moreover, measurements thought to be in LOS conditions can have sometimes bigger errors because of an unexpected reflection of the signal. Thus, detecting LOS or NLOS conditions is not always interesting in a UWB-based RTLS.

In order to put the focus on the identification of those measurements performed with a higher error, the Good/Medium/Bad classification model is proposed to be applied. As mentioned above, the steps followed to generate the model are the same as the previous model, obtaining the results described below.

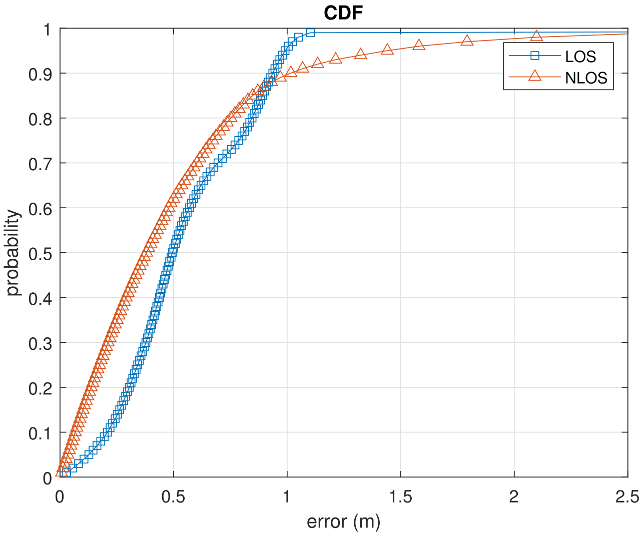

FIGURE 11. Cumulative distribution function (CDF) of the ranging errors separating LOS and NLOS measurements.

TABLE 7. LOS and NLOS error description.

	μ_r	σ_r	RMSE_r	ϵ_r $(P = 95\%)$	Max_r
LOS	0.558	0.445	0.714	0.991	9.295
NLOS	0.511	0.546	0.748	1.441	7.639

1) HYPERPARAMETER TUNING

Although the input data set is the same as in the previous model, in this case the feature to be predicted is different. Therefore, the new generated model varies, and with it the hyperparameters that generate it.

Figure 12 shows the F1 obtained by repeated cross-validation for the different numbers of trees that would form the model. In this case, selecting 250 trees, the F1 obtained converges, obtaining a maximum validation F1 of 0.74.

Once this value is fixed, a grid search of the rest of the hyper parameters is performed. Figure 13 shows the results obtained, where the values described in Table 8 are selected. Through these values, the final classification model is created using the total set of training data.

2) FEATURE SELECTION

In this model, each of the features will have the importance shown in Figure 14. In order to reduce the set of variables, keeping only those that really provide information, the feature selection process is repeated. During this phase, the least important features will be removed in each iteration, evaluating the quality of the created model with different amounts of variables.

Figure 15 shows the results obtained in the RFE. In this case, the best results are obtained with 13, 12 and 11 features. The difference of the obtained performance among these three cases is minimal, so the model can be done without the 2 less important features. These two features are FP_INDEX and STDEV_NOISE.

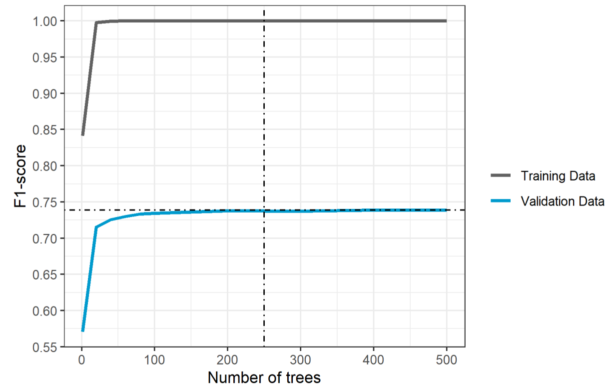

FIGURE 12. ***num trees*** **selection in Error classification.**

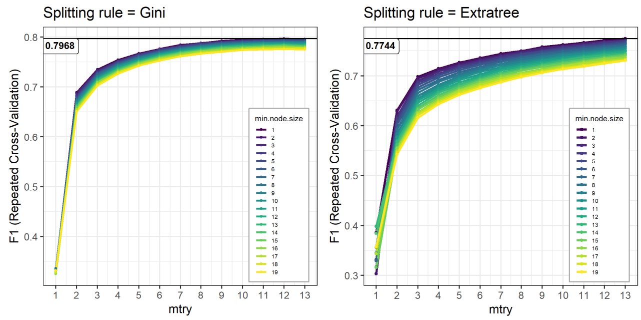

FIGURE 13. Hyperparameter tuning in Error classification.

TABLE 8. Selected hyperparameters in error classification.

	num trees	mtry	min node size	splitrule
Tuned value	250	12	1	Gini

3) MODEL PERFORMANCE

As with the previous model, the model has been evaluated using the test data set. In addition, the results using the total set of the variables and the reduced subset selected by feature selection have been compared. Table 9 and Table 10 show the confusion matrices with the total set of variables and with the feature selection respectively. In these tables, each cell represents how many samples have been classified in each class. The title of the column represents which class the samples belong to, whereas the rows tell which class they have been classified to. Taking these data, the Precision, Recall and F1-score are calculated as described in Section III-B and shown in Table 11.

Taking the results with the total set of features, it can be concluded that 87.9% of the test samples have been predicted correctly. The Bad class had the best performance, with an F1 score of 93.1%, followed by the Good class with 80.0% and the Medium class with 69.9%. The measures that may affect more significantly in the accuracy of the positioning system are the Bad measures that have been predicted as Good. In this case these measures are 0.9% of the total Bad samples, which is a really low value.

Using the reduced set of features, the correctly predicted measurements are increased to 88.1%. As before, the

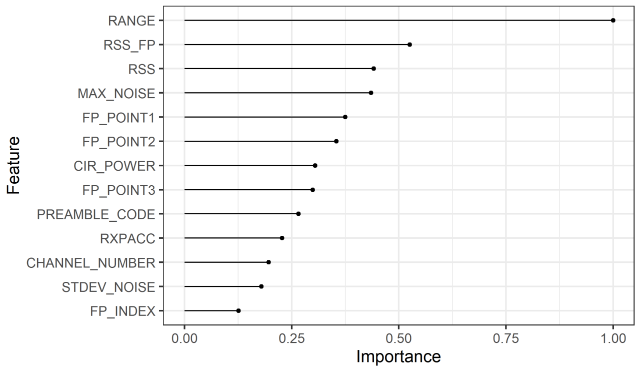

FIGURE 14. Feature importance in Error classification.

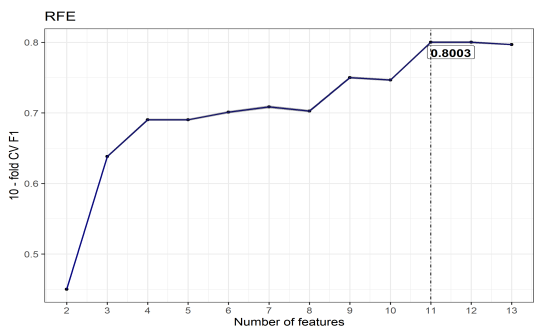

FIGURE 15. RFE Feature Selection in Error classification.

TABLE 9. Error classification model confusion matrix.

Prediction	Good	Medium	Bad
Good	16 207	1676	846
Medium	1672	14 824	1930
Bad	3898	7506	96 148

TABLE 10. Error classificacion model confusion matrix after feature selection.

Prediction	Good	Medium	Bad
Good	16 312	1643	864
Medium	1673	15 111	1978
Bad	3792	7252	96 082

TABLE 11. Error classification performance.

Metric	TF	FS
Accuracy	87.9%	88.1%
Precision	85.5%	85.6%
Recall	77.8%	78.3%
F1	81.0%	81.4%
Sample Prediction time	31.2 μs	29.9 μs

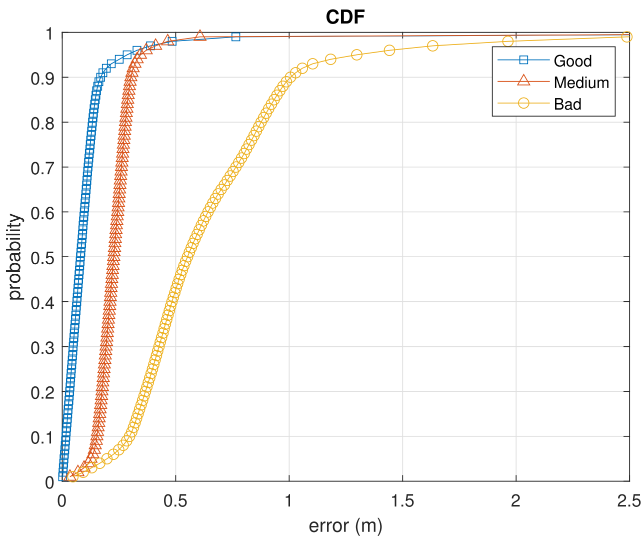

FIGURE 16. CDF of the ranging errors of measurements classified as Good, Medium and Bad.

measurements with a higher error that have been detected as Good represent only the 0.9% of the data.

As expected, the class that better predicts the model is the class of Bad measures. The reason for this is that the amount of measures available in this class is greater than the rest. As a result, the model possesses more information about the behavior of these measures type and therefore has a better knowledge of them. Thus, they are predicted more accurately.

All in all, eliminating the least relevant features has been beneficial for the classification model. All metrics are similar or slightly better. Moreover, the most important result is that the RFE reduces the sample prediction time without negatively affecting the classification performance. By reducing the sample prediction time from 31.2 μs to 29.9 μs, it is possible to predict 1393 more samples per second with the reduced model than with the one that uses the whole set of features. It is also important to remark that such a reduced sample prediction time allows the real-time applicability of the proposed model.

Focusing on the predictions obtained by the model, the aim is to analyze the measures that would be maintained once those predicted as Bad have been discarded. To this end, the results of the model obtained using the reduced set of features have been used, since it is the one with which the best results have been achieved.

Figure 16 shows the CDFs of the measurements classified as Good, Medium or Bad. Blue squares represent samples predicted as Good, orange triangles represent samples predicted as Medium and yellow circles represent samples predicted as Bad. Unlike Figure 11, each group of measurements has a different magnitude of ranging errors. Thus, it can be concluded that using an ML model to classify ranging estimates according to their error magnitude is feasible and effective.

For further analysis of the proposed classification model, Table 12 shows ranging error metrics of each group of measurements classified as Good, Medium or Bad. The format of the error metrics is the same as in Table 7. As it can be observed in Table 12, our proposed classification model, even with an accuracy of 88%, can effectively predict the accuracy of the ranging estimates. Those ranging estimates predicted as Good have an RMSE of 0.183 m, whereas those predicted as Medium 0.265 m. However, if measurements

TABLE 12. Error metrics of measurements predicted in each category.

	μ_r	σ_r	RMSE$_r$	ϵ_r ($P = 95\%$)	Max$_r$
Good	0.106	0.149	0.183	0.286	4.499
Medium	0.236	0.121	0.265	0.345	4.459
Bad	0.658	0.520	0.839	1.298	9.305

were classified as LOS or NLOS with an accuracy of 100% and those in LOS were selected, Table 7 shows that an RMSE of 0.714 m would have been obtained. Focusing on those measurements classified as Bad, the obtained RMSE is 0.839 m, bigger than the RMSE of 0.748 m that present those measurements in NLOS. Finally, it is remarkable that if those measurements predicted as Good are chosen, an error lower than 0.286 m is obtained in 95% of cases. This number is significantly better than the one obtained with LOS measurements, where an error lower than 0.991 m is obtained in 95% of cases, even if the LOS/NLOS classifier had an accuracy of 100%.

V. CONCLUSION

This paper has proposed two ML classification models to detect bad UWB ranging estimates using parameters given by the DWM1000 module of Qorvo-Decawave. The first proposed model classifies ranging samples as LOS or NLOS. The obtained classification results are better than those of the literature while guaranteeing the real-time applicability.

Although the proposed model that classifies ranging estimates as in LOS or NLOS performs better than the state of art, one of the main findings of this paper is that traditional LOS/NLOS classification approaches are not effective to select the best ranging estimates for an RTLS. Not all NLOS measurements give bad ranging estimates and many LOS measurements can be incorrectly labelled because of unexpected reflections. Using a public database, we have shown that classifying measurements as LOS and NLOS do not have significant difference in ranging error statistics.

If ranging estimates are to be classified in an RTLS, their ranging error should be estimated rather than the LOS or NLOS condition. The final position estimate provided by an RTLS is directly influenced by the ranging errors of UWB measurements. Thus, this paper has also proposed a classification model that estimates if received ranging estimates are Good, Medium or Bad according to the distance error in the ranging estimate and demonstrated its feasibility. High classification accuracy and F1 score have been obtained, with a sample prediction time of 29.9 μs, guaranteeing the real-time applicability of the proposed classification model. Moreover, if those measurements predicted as Good are to be used in an actual RTLS, a ranging RMSE of 0.183 m would have been obtained.

ACKNOWLEDGMENT

The authors would like to thank Luis Vitores Valcárcel García for his help in the revision of this manuscript and his improvement suggestions.

REFERENCES

[1] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, ''Increasing flexibility and productivity in industry 4.0 production networks with autonomous mobile robots and smart intralogistics,'' *Ann. Oper. Res.*, vol. 308, nos. 1–2, pp. 125–143, Jan. 2022.

[2] L. Zamora-Cadenas, I. Velez, and J. E. Sierra-Garcia, ''UWB-based safety system for autonomous guided vehicles without hardware on the infrastructure,'' *IEEE Access*, vol. 9, pp. 96430–96443, 2021.

[3] A. Ochoa-de-Eribe-Landaberea, L. Zamora-Cadenas, O. Peñagaricano-Muñoa, and I. Velez, ''UWB and IMU-based UAV's assistance system for autonomous landing on a platform,'' *Sensors*, vol. 22, no. 6, p. 2347, Mar. 2022.

[4] S. Krishnan and R. X. Mendoza Santos, ''Real-time asset tracking for smart manufacturing,'' in *Implementing Industry 4.0: The Model Factory as the Key Enabler for the Future of Manufacturing*, C. Toro, W. Wang, and H. Akhtar, Eds. Cham, Switzerland: Springer, 2021, pp. 25–53.

[5] J. Park, K. Kim, and Y. K. Cho, ''Framework of automated construction-safety monitoring using cloud-enabled BIM and BLE mobile tracking sensors,'' *J. Construction Eng. Manage.*, vol. 143, no. 2, Feb. 2017, Art. no. 05016019, doi: 10.1061/(ASCE)CO.1943-7862.0001223.

[6] Y. Huang, A. Hammad, and Z. Zhu, ''Providing proximity alerts to workers on construction sites using Bluetooth low energy RTLS,'' *Autom. Construction*, vol. 132, Dec. 2021, Art. no. 103928.

[7] C. D. Fernandes, A. Depari, E. Sisinni, P. Ferrari, A. Flammini, S. Rinaldi, and M. Pasetti, ''Hybrid indoor and outdoor localization for elderly care applications with LoRaWAN,'' in *Proc. IEEE Int. Symp. Med. Meas. Appl. (MeMeA)*, Jun. 2020, pp. 1–6.

[8] L. Nosrati, M. S. Fazel, and M. Ghavami, ''Improving indoor localization using mobile UWB sensor and deep neural networks,'' *IEEE Access*, vol. 10, pp. 20420–20431, 2022.

[9] H.-Y. Lin and M.-C. Yeh, ''Drift-free visual SLAM for mobile robot localization by integrating UWB technology,'' *IEEE Access*, vol. 10, pp. 93636–93645, 2022.

[10] R. J. Fontana, ''Recent system applications of short-pulse ultra-wideband (UWB) technology,'' *IEEE Trans. Microw. Theory Techn.*, vol. 52, no. 9, pp. 2087–2104, Sep. 2004.

[11] A. Liu, S. Lin, J. Wang, and X. Kong, ''A succinct method for non-line-of-sight mitigation for ultra-wideband indoor positioning system,'' *Sensors*, vol. 22, no. 21, p. 8247, Oct. 2022.

[12] X. Li, Y. Wang, and K. Khoshelham, ''A robust and adaptive complementary Kalman filter based on Mahalanobis distance for ultra wideband/inertial measurement unit fusion positioning,'' *Sensors*, vol. 18, no. 10, p. 3435, Oct. 2018.

[13] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, ''NLOS identification and mitigation for localization based on UWB experimental data,'' *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.

[14] W. Li, T. Zhang, and Q. Zhang, ''Experimental researches on an UWB NLOS identification method based on machine learning,'' in *Proc. 15th IEEE Int. Conf. Commun. Technol.*, Nov. 2013, pp. 473–477.

[15] C. Huang, A. F. Molisch, R. He, R. Wang, P. Tang, B. Ai, and Z. Zhong, ''Machine learning-enabled LOS/NLOS identification for MIMO systems in dynamic environments,'' *IEEE Trans. Wireless Commun.*, vol. 19, no. 6, pp. 3643–3657, Jun. 2020.

[16] S. Krishnan, R. X. M. Santos, E. R. Yap, and M. T. Zin, ''Improving UWB based indoor positioning in industrial environments through machine learning,'' in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1484–1488.

[17] R. Zandian and U. Witkowski, ''NLOS detection and mitigation in differential localization topologies based on UWB devices,'' in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2018, pp. 1–8.

[18] K. K. Cwalina, P. Rajchowski, O. Blaszkiewicz, A. Olejniczak, and J. Sadowski, ''Deep learning-based LOS and NLOS identification in wireless body area networks,'' *Sensors*, vol. 19, no. 19, p. 4229, Sep. 2019.

[19] M. Kolakowski and J. Modelski, ''Detection of direct path component absence in NLOS UWB channel,'' in *Proc. 22nd Int. Microw. Radar Conf. (MIKON)*, 2018, pp. 247–250.

[20] R. Bharadwaj, A. Alomainy, and S. K. Koul, ''Experimental investigation of body-centric indoor localization using compact wearable antennas and machine learning algorithms,'' *IEEE Trans. Antennas Propag.*, vol. 70, no. 2, pp. 1344–1354, Feb. 2022.

[21] V. Barral, C. J. Escudero, J. A. García-Naya, and P. Suárez-Casal, "Environmental cross-validation of NLOS machine learning classification/mitigation with low-cost UWB positioning systems," *Sensors*, vol. 19, no. 24, p. 5438, Dec. 2019.

[22] C. L. Sang, B. Steinhagen, J. D. Homburg, M. Adams, M. Hesse, and U. Rückert, "Identification of NLOS and multi-path conditions in UWB localization using machine learning methods," *Appl. Sci.*, vol. 10, no. 11, p. 3980, Jun. 2020.

[23] K. Bregar and M. Mohorcic, "Improving indoor localization using convolutional neural networks on computationally restricted devices," *IEEE Access*, vol. 6, pp. 17429–17441, 2018.

[24] Qorvo-Decawave. *DW1000 User Manual*. Accessed: Apr. 18, 2023. [Online]. Available: https://www.qorvo.com/products/p/DW1000

[25] A. Norrdine, "An algebraic solution to the multilateration problem," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, vol. 13, 2012, p. 15.

[26] T. Yang, A. Cabani, and H. Chafouk, "A survey of recent indoor localization scenarios and methodologies," *Sensors*, vol. 21, no. 23, p. 8086, Dec. 2021.

[27] W. Xu, W. Chen, and Y. Liang, "Feasibility study on the least square method for fitting non-Gaussian noise data," *Phys. A, Stat. Mech. Appl.*, vol. 492, pp. 1917–1930, Feb. 2018.

[28] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello, "Bayesian filtering for location estimation," *IEEE Pervasive Comput.*, vol. 2, no. 3, pp. 24–33, Jul. 2003.

[29] H. Brunner, M. Stocker, M. Schuh, M. Schuß, C. A. Boano, and K. Römer, "Understanding and mitigating the impact of Wi-Fi 6E interference on ultra-wideband communications and ranging," in *Proc. 21st ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, May 2022, pp. 92–104.

[30] S. Liu, H. Yang, Z. Mei, X. Xu, and Q. He, "Ultra-wideband high accuracy distance measurement based on hybrid compensation of temperature and distance error," *Measurement*, vol. 206, Jan. 2023, Art. no. 112276.

[31] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, pp. 5–32, Oct. 2001.

[32] P. Probst, M. N. Wright, and A. Boulesteix, "Hyperparameters and tuning strategies for random forest," *WIREs Data Mining Knowl. Discovery*, vol. 9, no. 3, Jan. 2019, Art. no. e1301, doi: 10.1002/widm.1301.

[33] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemometric Intell. Lab. Syst.*, vol. 83, no. 2, pp. 83–90, Sep. 2006.

[34] Ploomber. *Feature Selection*. Accessed: Jan. 12, 2024. [Online]. Available: https://sklearn-evaluation.ploomber.io/en/latest/optimization/feature_selection.html

[35] G. James, D. Witten, T. Hastie, and R. Tibshirani, "Linear model selection and regularization," in *An Introduction to Statistical Learning: With Applications in R*. New York, NY, USA: Springer, 2021, pp. 225–288.

[36] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo Method*, 3rd ed. Hoboken, NJ, USA: Wiley, 2017.

[37] C. Strobl, A.-L. Boulesteix, and T. Augustin, "Unbiased split selection for classification trees based on the Gini index," *Comput. Statist. Data Anal.*, vol. 52, no. 1, pp. 483–501, Sep. 2007.

MERITXELL ARSUAGA was born in San Sebastián, Spain. She received the B.Sc. degree in industrial technologies engineering from the University of Navarra, in 2015, and the M.Sc. degree in mathematical engineering from Complutense University, in 2017.

She has been a Research Staff Member with CEIT, San Sebastián, from 2019 to 2023. She is currently a Data Scientist with Ibermática an Ayesa Company. Her research interests include data analysis and visualization, machine learning techniques, and big data.

AITOR OCHOA-DE-ERIBE-LANDABEREA was born in San Sebastián, Spain, in 1995. He received the B.Sc. degree in mechanical engineering and M.Sc. degree in industrial engineering from the University of Navarra, in 2017 and 2019, respectively. He joined CEIT, in 2019, and works with the Group of Smart Systems for Industry 4.0 in support of his Ph.D. studies. His current research interests include localization systems, positioning algorithms, and sensor fusion.

LETICIA ZAMORA-CADENAS was born in León, Spain. She received the B.Sc. and M.Sc. degrees in telecommunications engineering from the University of Valladolid, Spain, in 2005 and 2008, respectively, and the Ph.D. degree in electrical and electronics engineering from Universidad de Navarra, in 2014.

She has been a Research Staff Member with CEIT, San Sebastian, Spain, since 2008, and a Lecturer with the University of Navarra, since 2014. Her research interests include localization systems, positioning algorithms and Bayesian filtering, and data fusion and signal processing for telecommunication systems.

SAIOA ARRIZABALAGA was born in Azkoitia, Spain, in 1979. She received the M.Sc. degree in telecommunication engineering from the University of the Basque Country, in 2003, and the Ph.D. degree from the University of Navarra, Spain, in 2009. She has been a Research Staff Member with CEIT, San Sebastian, Spain, since 2003, and an Associate Professor with the University of Navarra, since 2009. She is currently the Director of the Data Analysis and Information Management Research Group, CEIT. She has been involved in international and national projects related to positioning, 5G, and data analytics technologies. Her current research interests include cybersecurity and data analytics.

IGONE VELEZ was born in Mondragón, Spain, in 1977. She received the M.Sc. degree in electrical, electronic and control engineering and the Ph.D. degree from the University of Navarra, Spain, in 2000 and 2005, respectively.

She has been a Research Staff Member with CEIT, San Sebastian, Spain, since 2000, and an Associate Professor with the University of Navarra, since 2005. She is currently the Director of the ICT Division, CEIT. She has been involved in international and national projects related to UWB, RFID, and 5G technologies. Her current research interests include signal processing and hardware development for telecommunication systems and positioning systems.