UWB Positioning Integrity Estimation Using Ranging Residuals and ML Augmented Filtering

Mihkel Tommingas, Muhammad Mahtab Alam, *Senior Member, IEEE*, Ivo Müürsepp, and Sander Ulp

***Abstract*—This article investigates the use of ultrawideband (UWB) ranging residuals for coordinate integrity estimation and their use in a filtering scheme. Typically, UWB system accuracy is improved using channel statistics (CSs) to detect and mitigate non-line-of-sight effects between UWB sensors and the object to be located, potentially improving the end coordinate solution. However, in practice, when considering UWB system with a high positioning update rate, this is not a feasible approach, as gathering and processing CS data takes too much time. In contrast to this approach, this article proposes a set of features based on UWB ranging residuals that could be used as an alternative in integrity assessment. By using machine learning (ML), the most important features were extracted from the initial set, and then, used to train and validate a model for UWB coordinate error prediction. Finally, the prediction was applied in an adaptive Kalman filtering scheme as an input for measurement uncertainty. Model testing was done using UWB measurement test dataset gathered at an industrial site. The overall results showed significant improvement in 2-D and 3-D positioning metrics of ML-augmented filtering when compared to non-ML filtering. On average, the end coordinates in the test set had ca. 10 cm smaller mean location error and ca. 40 cm smaller dispersion in 2-D positioning. In addition, the presence of outliers was reduced significantly as the maximum error offset decreased by several meters. Although ML augmented filtering is computationally slower than non-ML filtering (e.g., ordinary and extended Kalman filter), it is still faster than using CS for UWB integrity estimation. The results show that using the proposed residual features in an ML model provides a feasible approach to predict UWB positioning integrity and use it as a measure of uncertainty in a coordinate filtering scheme.**

***Index Terms*—End coordinate correction and filtering, machine learning (ML), ranging residuals, ultrawideband (UWB) positioning.**

I. Introduction

CREATING an ultrawideband (UWB) positioning solution that provides reliable location information in a difficult industrial environment is a challenging task. For example, storage areas, whether indoor or outdoor, are usually littered with objects that obstruct radio frequency (RF) signal propagation, thus affecting the estimated coordinate of an object to be positioned.

Manuscript received 19 January 2024; revised 24 April 2024; accepted 18 June 2024. Date of publication 24 June 2024; date of current version 8 July 2024. This work was supported by the European Union's Horizon 2020 Research Programme under Grant "5G-TIMBER" agreement 101058505, in part by Just Transition Fund (JTF) project "Increasing the knowledge intensity of Ida-Viru entrepreneurship" (cofunded by the European Union), in part by the Estonian Research Council under Grant PUT-PRG424, and in part by the project of "Development of an industrial digital control system based on precise positioning technology ELIKO TAK and Atemix Automatika" under Grant 2014-2020.4.02.21-0311. *(Corresponding author: Mihkel Tommingas.)*

Mihkel Tommingas is with the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, 19086 Tallinn, Estonia, and also with the OÜ Eliko Tehnoloogia Arenduskeskus, 12918 Tallinn, Estonia (e-mail: mihkel.tommingas@taltech.ee).

Muhammad Mahtab Alam and Ivo Müürsepp are with the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, 19086 Tallinn, Estonia (e-mail: muhammad.alam@taltech.ee; ivo.muursepp@taltech.ee).

Sander Ulp is with the OÜ Eliko Tehnoloogia Arenduskeskus, 12918 Tallinn, Estonia (e-mail: sander.ulp@eliko.ee).

Digital Object Identifier 10.1109/JISPIN.2024.3418296

UWB-based systems are considered more robust in the presence of multipath effects and are less susceptible to interference as compared to other RF-signal-based positioning systems [1], [2], [3]. UWB system employs RF signals with a large bandwidth to be used in a wireless positioning scheme [4]. By using a two-way-ranging (TWR) approach, distances between fixed UWB nodes (anchors) and a mobile UWB node (tag) are measured and the final position of the tag is estimated based on these distances [2], [5]. Although a robust positioning solution, the ranging still relies on wireless RF signals. Thus, the accuracy and precision (i.e., integrity) of the end coordinate is affected but not limited by factors such as the number of servicing anchors, their vicinity to the tag, impairments caused by non-line-of-sight (NLOS) and suitable anchor layout geometry [1], [6], [7].

Problems related to NLOS detection and mitigation are an extensively researched topic in UWB-based positioning [5]. While the number of anchors and their spatial geometry can be adjusted according to the operating area, NLOS appears dynamically with constantly changing obstructions between the anchors and a moving tag. According to the works published by various authors, it can be seen that detection of NLOS and combating multipath effects is usually done by analyzing the characteristics of the RF propagation channel [i.e., channel state information (CSI)] [8], [9], [10], [11]. For example, channel impulse response (CIR), which describes the propagation path of a signal,

can be used to assess the amplitude and phase of a particular multipath component [12]. Although this information is effective for NLOS detection, then CIR entails also some constraints. Certain authors have noted that gathering CIR information requires a significant amount of data, hence causing a latency of approximately one second just to transfer the measurements [13]. Taking into account UWB positioning solution with a high position update rate, the transfer and processing of CIR information becomes unpractical. In addition, it is stated that CSI has to be collected for different types of environments, as a dataset describing a residential environment might not be suitable for a harsh industrial environment [5]. Finally, considering machine learning (ML)-based positioning algorithms that must be trained on real ranging or positioning data, using such an approach in conjunction with CSI increases computational complexity even further [2].

In essence, this article proposes positioning integrity assessment without the knowledge of CSI. In the literature, alternative methods have been used before. Barral et al. [13], [14] used received signal strength and range information for ML-based LOS/NLOS detection and classification. Liu et al. [15] investigated NLOS detection and mitigation using sum of squares (SSQ) of distance residuals. A large SSQ compared to a certain threshold would indicate inconsistency in localization. Similarly, Silva and Hancke [5] used SSQ of distance residuals for NLOS identification. A residual test was proposed by Chan et al. [16] in order to determine and identify the number of LOS base stations. However, this article expands on the analysis of ranging residuals further by adding features that describe their statistical and quantitative properties. In addition, aspects related to end coordinate calculations and geometrical dilution of precision (DOP) were also included. The goal was to include characteristics that describe UWB positioning integrity whether affected by LOS/NLOS or varying anchor geometry. Therefore, this article considers real-life measurement data that already contains both LOS and NLOS ranging measurements and it is assumed that end coordinate error is predicted regardless of tag's LOS/NLOS conditions or its position relative to anchors.

In contrast to most related works being done with simulated data, this article considers real-life measurements gathered from three different indoor environments. The calculated features from raw ranging data were then used in three distinct ML algorithms: regression tree (RT), random forest (RF), and XGBoost (XGB) [17], [18], [19]. These methods were used to produce three different models, which could estimate the offset from the true coordinate. An additional objective was to evaluate whether there was any significant gain to be had from using a more complex ML algorithm. Finally, the three different predictions were used as a measure of uncertainty in a coordinate filtering scheme in an adaptive Kalman filter (AKF), which was compared with the nonadaptive Kalman filter (KF) and extended Kalman filter (EKF).

The rest of this article is organized as follows. Section II describes the theory behind end coordinate estimation and ranging residuals. In addition, it is explained how residuals and their features are calculated. Section III gives an overview of data collection and processing with different ML algorithms along with coordinate filtering schemes. Section IV contains results by comparing the presented coordinate calculation methods. Finally, Section V concludes this article.

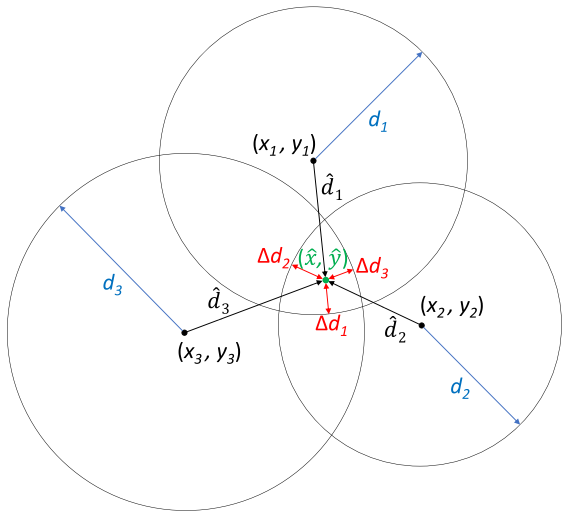


Fig. 1. Example of a 2-D trilateration scheme in UWB-based positioning with inaccurate range measurements. The difference between the distance to the estimated position $\hat{d}_i$ and the actual measured range d_i results in a residual Δd_i that can be used in estimating positioning integrity.

II. Coordinate Estimation Methods and Features

A. End Coordinate Estimation

Estimating the coordinates of the tag with regard to surrounding anchors presents a problem of multilateration. In Fig. 1, it can be seen how an object on coordinates $(\hat{x}, \hat{y})$ is located at certain distances from all surrounding anchors denoted with (x_i, y_i). By using known distance measurements d_i from each individual anchor, the tag's position can be estimated. Usually, for a single solution in 2-D space, at least three, and in 3-D space, four anchors are required [20]. It should also be considered that the anchors in Fig. 1 should not be positioned in a straight line as this may result in a flip ambiguity with possible solutions on either side of the line [21]. Under ideal conditions, without any measurement errors, $d_i = \hat{d}_i$ and the least-squares (LS) model provides a solution at the intersection of the three circles [22]. However, in real-life applications, ranging measurements contain errors caused by NLOS propagation and ranging noise, thus producing varying position estimates [2], [5].

B. End Coordinate Calculation

In this article, the end coordinate calculation of the tag is considered as a two-step process. First, estimating the initial position of the tag, and then, optimizing the solution with a nonlinear least-squares (NLS) approach. Both involve solving the multilateration problem using ranging measurements discussed in the previous section. In addition, this article considers

positioning in 3-D space. In step 1, a set of circle (1) is used to find the initial estimate of the tag's position $(\hat{x}, \hat{y}, \hat{z})$ as follows:

$$(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2 = d_i^2, \quad i = 1, 2, \ldots, N \tag{1}$$

where (x_i, y_i, z_i) is the known coordinate of the ith anchor and d_i is the measured distance between the tag and the ith anchor. The initial guess of the tag $(\hat{x}, \hat{y}, \hat{z})$ can be found by performing linearization on (1) and applying the LS method. First, an anchor (x_r, y_r, z_r) with the shortest measured distance to the tag d_r is taken as a reference point [23]. Next, the nonlinear expressions in all available circle equations N are expanded as

$$\begin{aligned} &x_i^2 - 2x_i\hat{x} + \hat{x}^2 + y_i^2 - 2y_i\hat{y} + \hat{y}^2 \\ &\quad + z_i^2 - 2z_i\hat{z} + \hat{z}^2 = d_i^2, \quad i = 1, 2, \ldots, N \end{aligned} \tag{2}$$

and the reference anchor (x_r, y_r, z_r) equation

$$\begin{aligned} &x_r^2 - 2x_r\hat{x} + \hat{x}^2 + y_r^2 - 2y_r\hat{y} + \hat{y}^2 \\ &\quad + z_r^2 - 2z_r\hat{z} + \hat{z}^2 = d_r^2 \end{aligned} \tag{3}$$

is subtracted from the rest of the expressions. The goal is to rearrange the terms with regards to unknowns $\hat{x}$, $\hat{y}$, and $\hat{z}$ in a way that satisfies the following linear model (4) as demonstrated by Guvenc, Chong, and Watanabe [24]:

$$A\theta = b \tag{4}$$

where

$$A = -2\begin{bmatrix} x_1 - x_r & y_1 - y_r & z_1 - z_r \\ x_2 - x_r & y_2 - y_r & z_2 - z_r \\ \vdots & \vdots & \vdots \\ x_{N-1} - x_r & y_{N-1} - y_r & z_{N-1} - z_r \end{bmatrix} \tag{5}$$

$$\theta = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix} \tag{6}$$

and

$$b = \begin{bmatrix} d_1^2 - d_r^2 - x_1^2 + x_r^2 - y_1^2 + y_r^2 - z_1^2 + z_r^2 \\ d_2^2 - d_r^2 - x_2^2 + x_r^2 - y_2^2 + y_r^2 - z_2^2 + z_r^2 \\ \vdots \\ d_{N-1}^2 - d_r^2 - x_{N-1}^2 + x_r^2 - y_{N-1}^2 + y_r^2 - z_{N-1}^2 + z_r^2 \end{bmatrix}. \tag{7}$$

Finally, the estimated tag's position θ has the following LS solution:

$$\theta = (A^T A)^{-1} A^T b. \tag{8}$$

As shown in an example in Fig. 1, it can be seen that the tag is estimated somewhere within the area overlapped by three circles. The sum of the squares of distance errors can further be minimized using the NLS approach [25]. As a step 2 of position estimation, the Gauss–Newton optimization algorithm was applied. The LS solution provides an initial estimate, along with previously measured anchor coordinates and individual distances measured between anchors and tag. The estimated position is found by minimizing the objective function

$$\begin{aligned} \hat{x}, \hat{y}, \hat{z} = \text{argmin}_{x,y,z} \sum_{i=1}^{N} \Big(&(x_i - x)^2 \\ &+ (y_i - y)^2 + (z_i - z)^2 - d_i^2 \Big)^2 \end{aligned} \tag{9}$$

where x, y, and z represent the coordinates that provide the smallest error. Since there are various methods to solve this nonlinear multilateration problem, this article applies linearization using Taylor series with the Gauss–Newton iteration procedure. Renaming the initial guess from the LS solution (8) as (x_G, y_G, z_G), the measured distances d_i are approximated through first-order Taylor series expansion as demonstrated by Guillory, Truong, and Wallerand [20] as follows:

$$\begin{aligned} &d_i(\hat{x}, \hat{y}, \hat{z}) \\ &\approx d_i(x_G, y_G, z_G) + \frac{\partial d_i(\hat{x}, \hat{y}, \hat{z})}{\partial \hat{x}}\Big|_{x_G, y_G, z_G} \Delta x \\ &\quad + \frac{\partial d_i(\hat{x}, \hat{y}, \hat{z})}{\partial \hat{y}}\Big|_{x_G, y_G, z_G} \Delta y + \frac{\partial d_i(\hat{x}, \hat{y}, \hat{z})}{\partial \hat{z}}\Big|_{x_G, y_G, z_G} \Delta z \\ &\approx d_i(x_G, y_G, z_G) \\ &\quad + \frac{\hat{x} - x_i}{\sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2}}\Big|_{x_G, y_G, z_G} \Delta x \\ &\quad + \frac{\hat{y} - y_i}{\sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2}}\Big|_{x_G, y_G, z_G} \Delta y \\ &\quad + \frac{\hat{z} - z_i}{\sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2 + (z_i - \hat{z})^2}}\Big|_{x_G, y_G, z_G} \Delta z \\ &\approx d_i(x_G, y_G, z_G) + \frac{\hat{x} - x_i}{d_i(\hat{x}, \hat{y}, \hat{z})}\Big|_{x_G, y_G, z_G} \Delta x \\ &\quad + \frac{\hat{y} - y_i}{d_i(\hat{x}, \hat{y}, \hat{z})}\Big|_{x_G, y_G, z_G} \Delta y + \frac{\hat{z} - z_i}{d_i(\hat{x}, \hat{y}, \hat{z})}\Big|_{x_G, y_G, z_G} \Delta z \\ &\approx d_i(x_G, y_G, z_G) + \frac{x_G - x_i}{d_i(x_G, y_G, z_G)} \Delta x \\ &\quad + \frac{y_G - y_i}{d_i(x_G, y_G, z_G)} \Delta y + \frac{z_G - z_i}{d_i(x_G, y_G, z_G)} \Delta z \end{aligned} \tag{10}$$

where Δx, Δy, and Δz are equal to $\hat{x} - x_G$, $\hat{y} - y_G$, and $\hat{z} - z_G$, respectively. Considering that Δx, Δy, and Δz are multiplied to first-order derivatives when

$$J_i = \begin{bmatrix} \frac{x_G - x_i}{d_i(x_G, y_G, z_G)} & \frac{y_G - y_i}{d_i(x_G, y_G, z_G)} & \frac{z_G - z_i}{d_i(x_G, y_G, z_G)} \end{bmatrix} \tag{11}$$

then (10) can be rearranged into matrix form as

$$\Delta d_{NLS} = J \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{12}$$

with Δd_{NLS} representing the difference between measured and estimated distances. The error corrections Δx, Δy, and Δz can be found by solving the normal equation as shown in (8) and substituting values accordingly

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = (J^T J)^{-1} J^T \Delta d_{\text{NLS}}. \tag{13}$$

Using the error correction vector, the initial guess coordinates x_G, y_G, and z_G are updated with Gauss–Newton iteration until a convergence criterion has been reached (e.g., until the error correction vector is sufficiently small [20]). After reaching a predetermined threshold, the final position estimation results as

$$\begin{bmatrix} x_G + \Delta x \\ y_G + \Delta y \\ z_G + \Delta z \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}. \tag{14}$$

Similarly to Fig. 1, it can be seen that the difference between an individual measured distance d_i, and distance $\hat{d}_i$ calculated from the estimated coordinate $(\hat{x}, \hat{y}, \hat{z})$, results in a residual Δd_i as

$$\Delta d_i = d_i - \hat{d}_i. \tag{15}$$

C. Features

This paragraph describes features used in the ML model training. As stated in the introduction, this article considers both previously used features in the literature as well as several novel ones. As ranging residuals could indirectly reflect the end coordinate integrity, several statistical metrics such as the residual mean or sample variance have been added. Additional features have also been included regarding end coordinate calculations and positioning geometry.

1) Residual Statistics: Depending on the location of the estimated solution, $\hat{d}_i$ may be longer or shorter compared to the individual measured distance d_i resulting in a positive or negative residual. A significant change in the magnitude of a residual may indicate that UWB propagation path is affected by an obstruction. Therefore, residual statistics were calculated for three different sets: positive, negative, and overall residuals. In addition, statistical equations were averaged to remove the dependence on the size of available residuals. The following statistics were calculated:

average SSQ

$$\text{SSQ} = \frac{\sum_{i=1}^{n} \Delta d_i^2}{n} \tag{16}$$

root mean square (RMS)

$$\text{RMS} = \sqrt{\frac{\text{SSQ}}{n}} \tag{17}$$

mean

$$\bar{x} = \frac{\sum_{i=1}^{n} \Delta d_i}{n} \tag{18}$$

mean absolute deviation (MAD)

$$\text{MAD} = \frac{\sum_{i=1}^{n} |\Delta d_i - \bar{x}|}{n} \tag{19}$$

standard deviation

$$s = \sqrt{\frac{\sum_{i=1}^{n} (\Delta d_i - \bar{x})^2}{n}} \tag{20}$$

and variance

$$v = s^2 \tag{21}$$

where n represents the number of residuals in a corresponding positive, negative, or overall set (also used as a feature).

2) LS and NLS Metrics: These values are associated with position calculation as discussed in Section II-B. The chosen parameters include Euclidean distance ΔD between LS (8) and NLS (14) solutions and the number of Gauss–Newton iterations to convergence n_{GN}. For the latter, there is no implicit equation as the iteration counter is initialized at each coordinate optimization process

$$\Delta D = \sqrt{(x_G - \hat{x})^2 + (y_G - \hat{y})^2 + (z_G - \hat{z})^2}. \tag{22}$$

3) Geometrical Integrity of Positioning: In an indoor positioning system, DOP indicates geometric location distribution. It contains the knowledge of positioning accuracy under specific base station network and scene characteristics [6]. Using the estimated coordinates of the tag, the DOP parameter indirectly shows the level of geometrical uncertainty in an area relative to the anchors. In this article, the position dilution of precision (PDOP) was used as it depends on x-, y-, and z-coordinates. To calculate PDOP, the set of ranging (1) can be implemented with precalculated end coordinates from (14). By finding partial derivatives with respect to each coordinate similarly as was shown in (10), the result is formulated in matrix form as

$$A_p = \begin{bmatrix} \frac{x_1 - \hat{x}}{d_1} & \frac{y_1 - \hat{y}}{d_1} & \frac{z_1 - \hat{z}}{d_1} & 1 \\ \frac{x_2 - \hat{x}}{d_2} & \frac{y_1 - \hat{y}}{d_2} & \frac{z_1 - \hat{z}}{d_1} & 1 \\ \vdots & \vdots & & \\ \frac{x_N - \hat{x}}{d_N} & \frac{y_N - \hat{y}}{d_N} & \frac{z_N - \hat{z}}{d_N} & 1 \end{bmatrix}. \tag{23}$$

Next, the covariance matrix Q is calculated from the LS normal matrix

$$Q = (A_p^T A_p)^{-1} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_y^2 & \sigma_{yz} \\ \sigma_{zx} & \sigma_{zy} & \sigma_z^2 \end{bmatrix}. \tag{24}$$

Fig. 2. Industrial site at Krah Pipes OÜ, which manufactures thermoplastic pipes in a complex industrial environment. The site contained a network of eight UWB anchors (layout in Fig. 19).

Finally, PDOP is calculated from trace of matrix Q as

$$\text{PDOP} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}. \tag{25}$$

III. Data Processing, Model Training, and Filtering

A. Data Collection

UWB measurement data were collected at three different sites: Krah Pipes OÜ factory, Eliko office, and Auroom Kastre factory, which all contained a set network of UWB sensors. Data from the first two sites were used to cross validate and train the ML model, while data from the third site were used for testing. Krah Pipes OÜ (see Fig. 2) is a company that manufactures thermoplastic pipes and in terms of RF propagation, presents a complex environment with constantly moving objects [26]. The Eliko real-time locating system (RTLS) was installed inside the manufacturing facility by placing UWB anchors at fixed locations while the tag was sequentially placed at different locations on the factory floor during the measurements. Based on the Qorvo's DW1000 chip, the RTLS was set to operate on UWB channel 4 [27]. Eliko RTLS also uses active–passive TWR protocol with clock offset error mitigation [28]. Ground-truth coordinates were measured in a local frame of reference with the Leica DISTO S910 measurement tool and assigned to eight UWB anchors, as well as 30 different tag locations around the facility. The measurement tool has an accuracy of ± 1 mm [29].

The measurement tool was positioned on a mezzanine floor in order to have LOS with all measurement points. By using an update rate of 10 Hz, each location was measured for 30 s, resulting in approximately 300 ranging sequences per location. A similar measurement procedure was performed at the Eliko office (30 measurement points) and Auroom Kastre factory (40 measurement points) using UWB positioning network of 17 and 15 anchors, respectively. The office environment provided additional training data in terms of poor PDOP conditions, i.e., measurements that were taken outside of the convex hull of the UWB anchor layout as can be seen in Fig. 18 (e.g., points 13, 14, and 15).

B. Data Processing and Model Training

Raw ranging data, collected during the measurements, was assembled into training-validation (Krah factory and Eliko office) and testing (Auroom factory) datasets containing end coordinates, true distances (dependent), and features (independent) described in Section II-C. It should be noted that data from the Auroom factory were not used in training in order to have a stand-alone dataset to test the general model. The purpose of the model was to predict end-coordinate error or offset based on precalculated independent features. After data cleaning and shuffling, the datasets were changed into the appropriate format for cross validation and training. Three ML methods were chosen: extreme gradient boosting, RT, and RF. The idea was to compare the prediction performance of a simple ML method (i.e., a single RT) against more complex ones.

The aforementioned ML algorithms and datasets were used in the R Studio environment [30]. For each ML method, tenfold cross validation was carried out to select hyperparameters that provide the smallest prediction error against the validation set. Essentially, the training dataset was separated into ten segments with one segment being the validation set. Such an approach helps to generalize the model and mitigate overfitting. Next, using chosen hyperparameters in an initial model, combination of most important features were selected for the final model.

R Studio provides appropriate cross-validation $train$ and $trainControl$ functions through the $caret$ library [31]. The main hyperparameters used were: tree depth and number of boosting iterations for XGB; tree depth and complexity parameter (CP) for RT and tree depth for RF.

It should be noted that no prior feature selection before model cross validation was done. Rather RT, RF, and XGB libraries in R Studio already contain built-in functions to output features that contribute the most in making the prediction.

1) RT Feature Selection and Training: In contrast to using a decision tree for classification task, end-coordinate offset is considered as a continuous target variable that is predicted using an RT. It is generated using a set of training samples with the corresponding response variables. A trained tree structure is then used to predict the value of an unknown test sample. It consists of root, branches, nodes, and leaves. Each internal node represents a feature, branches represent the feature values and leaf nodes represent the outcome of prediction [32].

Cross validation compared sets of training data using different RT depths in terms of prediction error. As can be seen in Fig. 3, a tree with a depth of 7 is sufficient for providing the least amount of error as choosing a deeper tree results in no further error mitigation. In addition, the tree can be pruned or optimized using a CP, which is the minimum improvement in the model needed at each node. CP is used to select the optimal size for the tree. As can be seen in Fig. 4, a tree size of 7 has a CP of 0.025, which outputs a tree shown in Fig. 5.

It should be noted that the $rpart$ library also removes surrogate features, i.e., features that provide the same goodness of split. Therefore, the final tree may have a different depth compared to cross-validated trees. Features were extracted by using a built-in $rpart.plot$ function and feature importance was

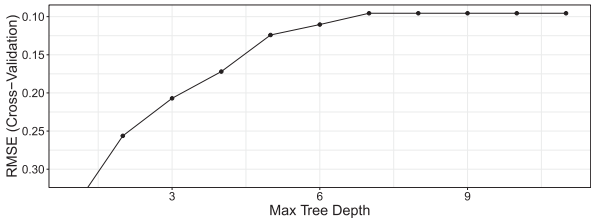


Fig. 3. Determining RT depth after tenfold cross validation. It can be seen that a tree depth of seven is enough to provide the smallest RMSE.

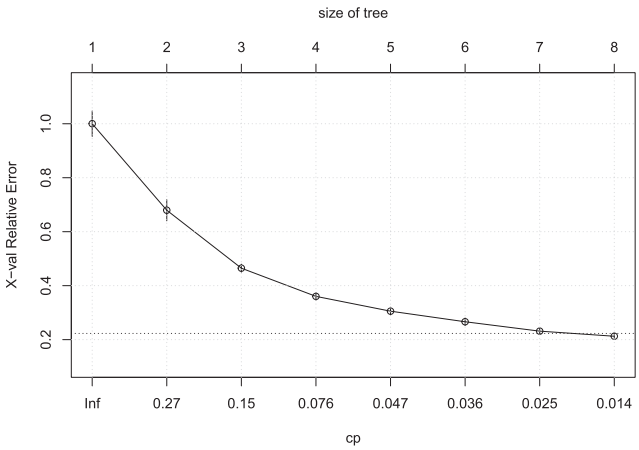


Fig. 4. After tenfold cross validation, a tree depth of seven corresponds to a CP of 0.025. These hyperparameters were used to generate the final RT (shown in Fig. 5).

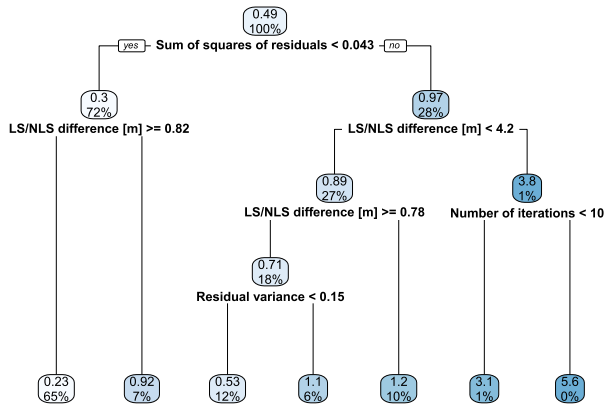


Fig. 5. Final RT to be used in ML prediction.

based on the goodness of split [17]. Features for the final model can be seen in Fig. 6.

2) RF Feature Selection and Training: In ensemble learning, bagging and boosting are two main approaches. RF can be viewed as an evolution of bagging methodology and can be used in classification and regression problems. It is defined as an ensemble of decision trees that implements randomness in the model-building process of each decision tree [33]. It can process

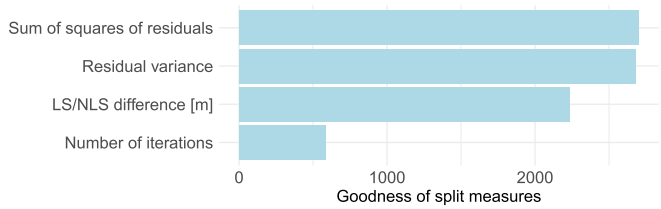


Fig. 6. Features used in the final RT model, which are ordered based on the goodness of split in an RT.

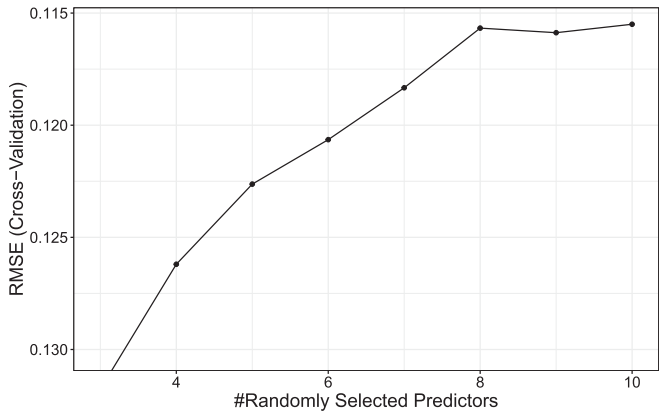


Fig. 7. Finding the optimal number of randomly selected RF predictors using cross validation with 100 random trees. It can be seen that using more than eight features results in no significant increase in cross-validation error.

high-dimensional data effectively, so it is different from neural networks. In RF, each tree acts as an independent regression function, and RTs are trained using different bootstrap samples of the training data. The average prediction of each individual tree is used as the final output [34].

RF training, validation, and testing were done using the *ranger* package, which is a fast implementation of RF suited for high-dimensional data [18]. Cross validation on training data showed how different number of RF predictors compare in terms of prediction error. As shown in Fig. 7, using 100 random trees with eight predictors provides a sufficient amount of error as using more than eight might lead to model overfitting and results in no significant reduction in RMSE. Next, feature selection was done for the initial RF model, with eight random predictors. By comparing different combinations of features, those with the least amount of error in predicting validation set response values were selected. As can be seen in Fig. 8, choosing more than seven features results in no significant decrease in prediction error. These features are presented in Fig. 9.

3) XGB Feature Selection and Training: On the other hand, XGB represents the approach of boosted ensemble learning. It is a large-scale general-purpose gradient boosting library, which has been seen to dominate structured and tabular datasets on classification, regression, and predictive modeling problems [19], [35]. The algorithm creates a sequential ensemble of tree models, all of which work to improve each other. The final prediction results in a summation of the predictions of multiple RTs. The

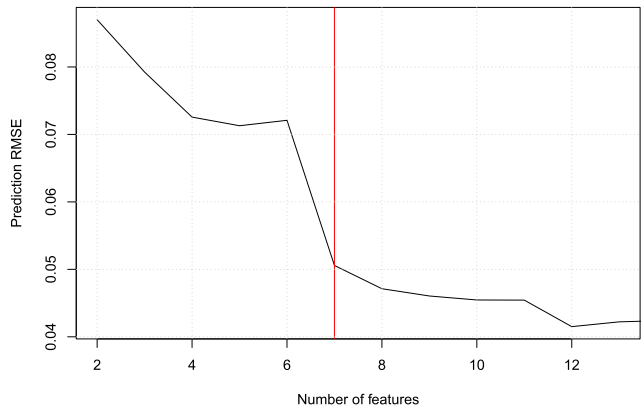


Fig. 8. Using more than seven most important features results in no significant RMSE decrease in RF prediction and could overfit the model.

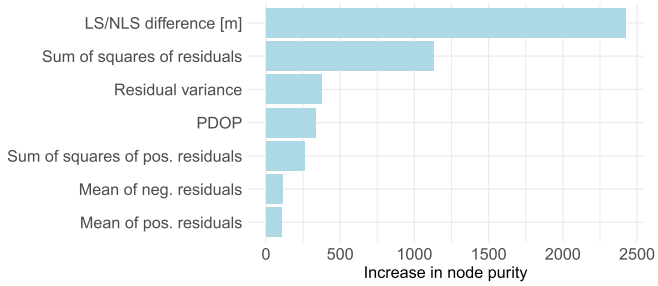


Fig. 9. Set of features in the final model that provide the biggest node purity increase in RF prediction.

XGB algorithm comprises a series of base classifiers such as decision tree, k-nearest neighbors, support vector machine, and logistic regression. These are linearly superimposed so that they work together to optimize the algorithm [36].

Using cross validation with $xgboost$ library, different sets of XGB parameters were compared in terms of prediction error as shown in Fig. 10. It can be seen that a model with a tree depth of 5 and 150 boosting iterations is sufficient as choosing more than 150 iterations would present no significant increase in prediction performance. In addition, feature selection was done using the initial model with aforementioned hyperparameters. By comparing different combinations of features, those with the least amount of error in predicting validation set response values were selected. As can be seen in Fig. 11, more than eight features provide only a marginal increase in predicting validation set response values. List of features used in the final XGB model is presented in Fig. 12.

C. Coordinate Filtering

As a final step, the end coordinate is estimated using a KF. In the current context, the filter averages end coordinates, while considering the uncertainty of measurements (prediction) and previously filtered coordinates. While in a traditional KF, the process and measurement noise have fixed values, then in

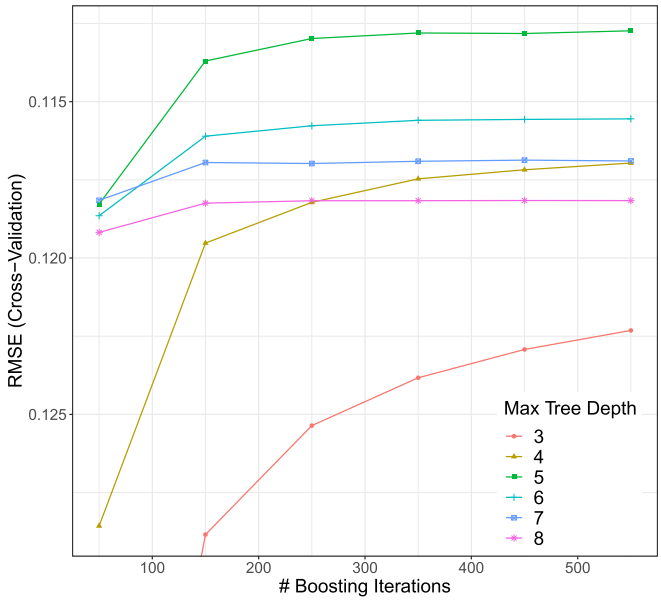


Fig. 10. Determining suitable XGB tree depth and number of boosting iterations using cross validation. Tree depth 5 and 150 boosting iterations are chosen parameters for the model. Choosing a higher number of iterations results in no significant decrease in RMSE and might lead to overfitting.

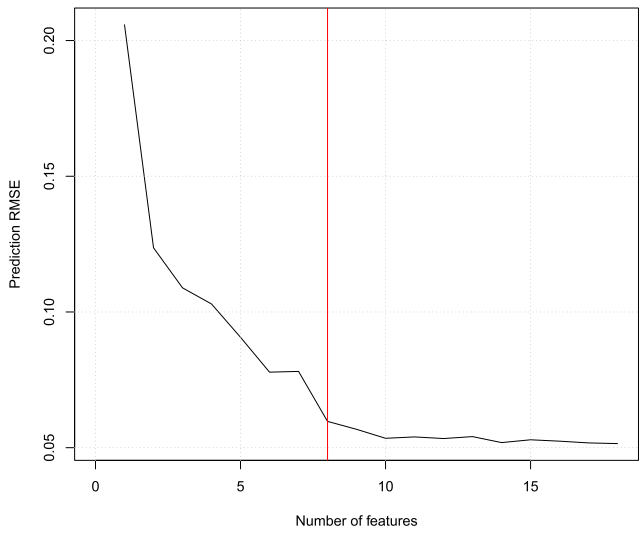


Fig. 11. Using more than eight features has no significant impact on XGB prediction accuracy.

real-life applications, it can be seen that measurement uncertainty is a dynamic value, which in turn is affected by external factors such as NLOS. Therefore, it is preferable to know the measurement uncertainty at every ranging calculation in order to estimate whether the current coordinate can be trusted or not.

In this article, the ML model predicts end coordinate offset from true value, based on features used in the ML model. Since, the direction of the error with regards to x-, y-, and z-axes is

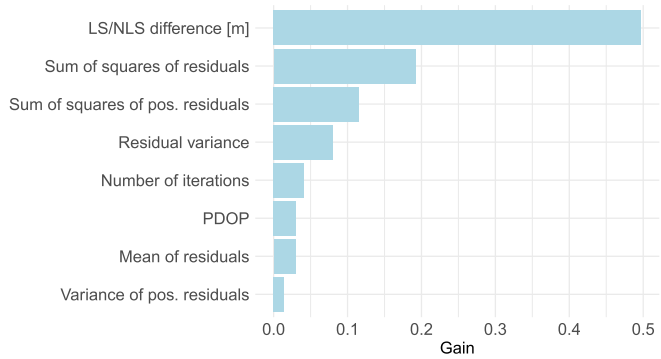


Fig. 12. Features used in the final XGB model.

Algorithm 1: Kalman Filter.

Input: $\hat{\mathbf{X}}_0, \mathbf{Z}_k, \mathbf{P}_0, \mathbf{Q}, \mathbf{R}$
Output: $\hat{\mathbf{X}}_k$
Initialize $\mathbf{A}, \mathbf{P}_0, \mathbf{H}, \mathbf{I}$
Prediction step
for $k = 1, \ldots, \infty$
1: State prediction $\hat{\mathbf{X}}_k^- = \mathbf{A}\hat{\mathbf{X}}_{k-1}$
2: Covariance prediction $\mathbf{P}_k^- = \mathbf{A}\mathbf{P}_{k-1}\mathbf{A}^T + \mathbf{Q}$
Correction step
3: Kalman gain $\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}$
4: State correction $\hat{\mathbf{X}}_k = \hat{\mathbf{X}}_k^- + \mathbf{K}_k(\mathbf{Z}_k - \mathbf{H}_k\hat{\mathbf{X}}_k^-)$
5: Covariance correction $\mathbf{P}_k = \mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)$
return $\hat{\mathbf{X}}_\mathbf{k}, \mathbf{P}_k$
end for

not known, this prediction can be considered as a measure of uncertainty in all three axes. By implementing the prediction as a dynamic measurement uncertainty in an AKF, positioning accuracy can be improved further.

1) KF and AKF Filtering: In this work, the main difference between a KF and AKF is in the application of the R matrix, which represents positioning measurement uncertainty. In KF, the diagonal elements of R in (26) were chosen as fixed values diag(0.01, 0.01, 0.01) corresponding to the precision of the DW1000 device [37] with

$$R_{\text{KF}} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix} = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}. \quad (26)$$

However, AKF measurement uncertainty in (27) is updated at each iteration as the end coordinate is calculated and ML prediction is added to the variance of diagonal elements as

$$R_{\text{AKF}} = \begin{bmatrix} 0.01 + \hat{D}_{\text{ML}} & 0 & 0 \\ 0 & 0.01 + \hat{D}_{\text{ML}} & 0 \\ 0 & 0 & 0.01 + \hat{D}_{\text{ML}} \end{bmatrix}. \quad (27)$$

In essence, the ML prediction drives the filtering process by dynamically changing measurement uncertainty, i.e., whether to trust measurement or process. In KF, EKF, and AKF, the process noise matrix Q has constant values diag(0.01, 0.01, 0.01). As shown at the beginning of Algorithm 1, the state transition matrix A, state covariance P_0, and observation matrix H are initialized as 3-by-3 identity matrices. $\hat{X}_0$ represents the first converged NLS solution from (14), Z_k is the measurement vector, and I is a 3-by-3 identity matrix.

2) EKF Filtering: Finally, ML-driven AKF is compared with the EKF, which is capable of dealing with nonlinear problems such as multilateration described in Section II-A. In contrast to KF and AKF, which predict and correct coordinates, EKF makes state corrections using residuals between measured distances Z_k and distances to the last estimated coordinates. In Algorithm 1, state correction step $H_k\hat{X}_k^-$ is replaced with D_k^- where

$$D_k^- = \begin{bmatrix} \sqrt{(x_k^- - x_1)^2 + (y_k^- - y_1)^2 + (z_k^- - z_1)^2} \\ \sqrt{(x_k^- - x_2)^2 + (y_k^- - y_2)^2 + (z_k^- - z_2)^2} \\ \vdots \\ \sqrt{(x_k^- - x_n)^2 + (y_k^- - y_n)^2 + (z_k^- - z_n)^2} \end{bmatrix} \quad (28)$$

with x_k^-, y_k^-, and z_k^- representing coordinates from last iteration. Measurement vector Z_k represents current iteration distance equations with added measurement noise

$$Z_k = \begin{bmatrix} \sqrt{(x_k - x_1)^2 + (y_k - y_1)^2 + (z_k - z_1)^2} + v_1 \\ \sqrt{(x_k - x_2)^2 + (y_k - y_2)^2 + (z_k - z_2)^2} + v_2 \\ \vdots \\ \sqrt{(x_k - x_n)^2 + (y_k - y_n)^2 + (z_k - z_n)^2} + v_n \end{bmatrix} \quad (29)$$

where v_k represents measurement noise vector, which has covariance matrix R_k as diag(0.01, 0.01, 0.01). Process noise matrix Q is also set as diag(0.01, 0.01, 0.01).

With EKF, the entire NLS approximation process discussed in Section II-B may be bypassed and do linearization through the observation matrix H, which is comprised of first-order partial derivatives as demonstrated by Kim et al. [37]

$$H_k = \begin{bmatrix} \frac{\partial d_1(x_k,y_k,z_k)}{\partial x_k} & \frac{\partial d_1(x_k,y_k,z_k)}{\partial y_k} & \frac{\partial d_1(x_k,y_k,z_k)}{\partial z_k} \\ \frac{\partial d_2(x_k,y_k,z_k)}{\partial x_k} & \frac{\partial d_2(x_k,y_k,z_k)}{\partial y_k} & \frac{\partial d_2(x_k,y_k,z_k)}{\partial z_k} \\ \vdots & \vdots & \vdots \\ \frac{\partial d_n(x_k,y_k,z_k)}{\partial x_k} & \frac{\partial d_n(x_k,y_k,z_k)}{\partial y_k} & \frac{\partial d_n(x_k,y_k,z_k)}{\partial z_k} \end{bmatrix} \quad (30)$$

where derivatives correspond to

$$\frac{\partial d_i(x_k,y_k,z_k)}{\partial x_k} = \frac{x_k - x_i}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2}} \quad (31)$$

$$\frac{\partial d_i(x_k,y_k,z_k)}{\partial y_k} = \frac{y_k - y_i}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2}} \quad (32)$$

$$\frac{\partial d_i(x_k,y_k,z_k)}{\partial z_k} = \frac{z_k - z_i}{\sqrt{(x_k - x_i)^2 + (y_k - y_i)^2 + (z_k - z_i)^2}}. \quad (33)$$

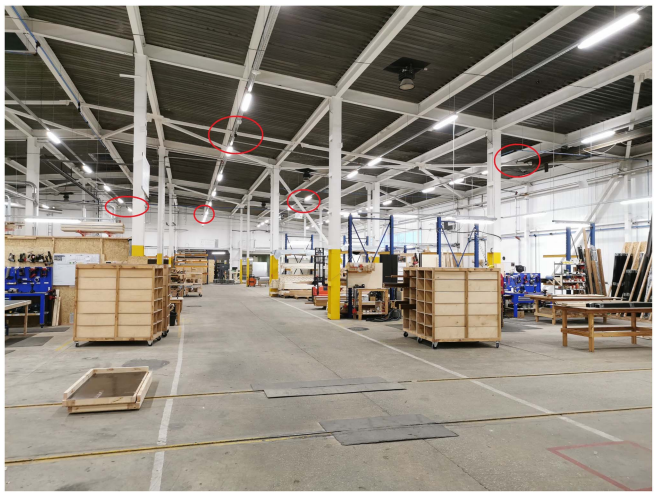


Fig. 13. Manufacturing area inside Auroom Kastre factory. Red circles highlight visible UWB anchors. Anchor layout can be seen on Fig. 17.

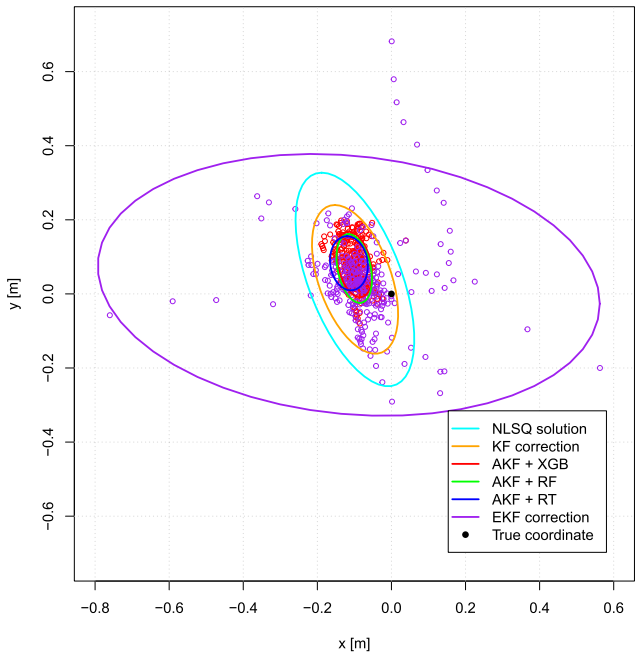


Fig. 14. Comparison of confidence ellipses for pt. 20 with respect to x- and y-axes. Each ellipse contains samples within one standard deviation (68% confidence). For the sake of clarity, only two point clouds are shown (EKF and AKF + XGB). It can be seen how prediction keeps the point cloud more tightly together, whereas the EKF relies only on noisy ranging data that produce much more sparsely distributed samples. Due to outliers, the figure has been zoomed in on the largest ellipse.

In the context of coordinate calculation, skipping the NLS coordinate calculations makes EKF computationally less demanding. On the other hand, a poor LS coordinate in the state vector can affect the filtering process and result in an inaccurate coordinate. Therefore, in this work for comparison purposes, an EKF was provided with a converged NLS coordinate as the initial state vector.

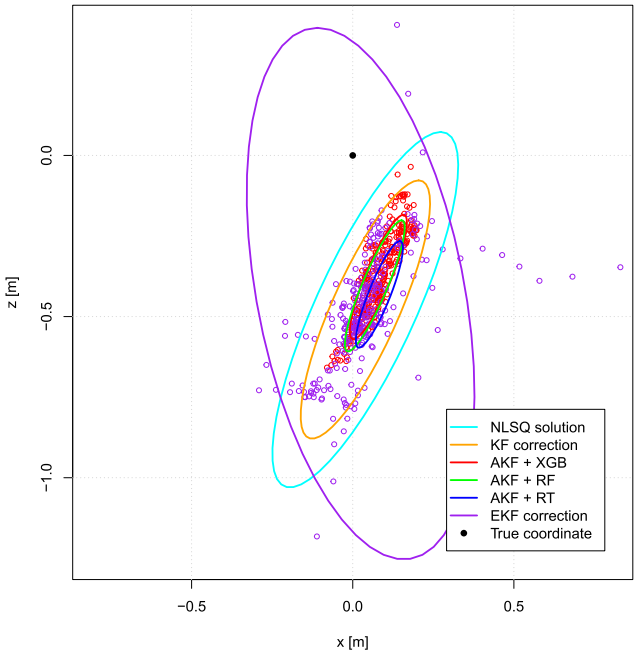


Fig. 15. Comparison of confidence ellipses for pt. 20 with respect to x- and z-axes. For the sake of clarity, only two point clouds are shown (EKF and AKF+XGB). In addition, due to outliers, the figure has been zoomed in on the largest ellipse.

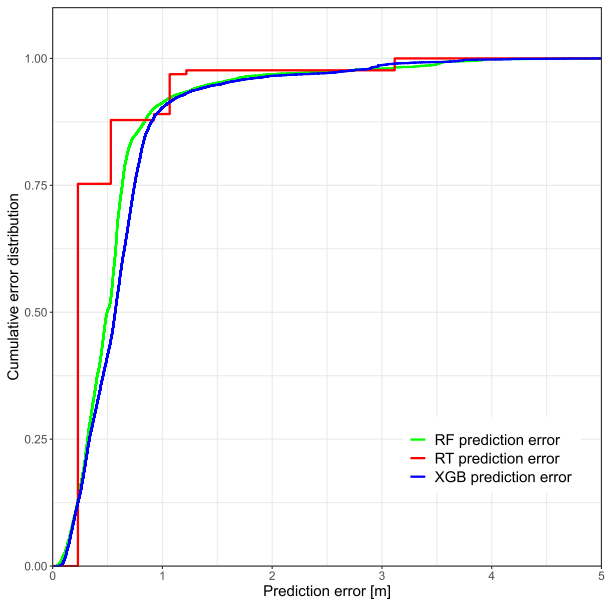


Fig. 16. Cumulative distribution of prediction errors. It can be seen that an RT provides more robust prediction levels according to leaf nodes from a single tree as shown in Fig. 5. RF and XGB predictions are smoother at the cost of more complex models.

IV. Results

Test data were measured in an industrial site at Auroom Kastre factory, which manufactures sauna modules as shown in Fig. 13. The measurement setup was similar to the Eliko office and Krah Pipes factory with 40 different measurement points scattered over the factory area as can be seen in Fig. 17.

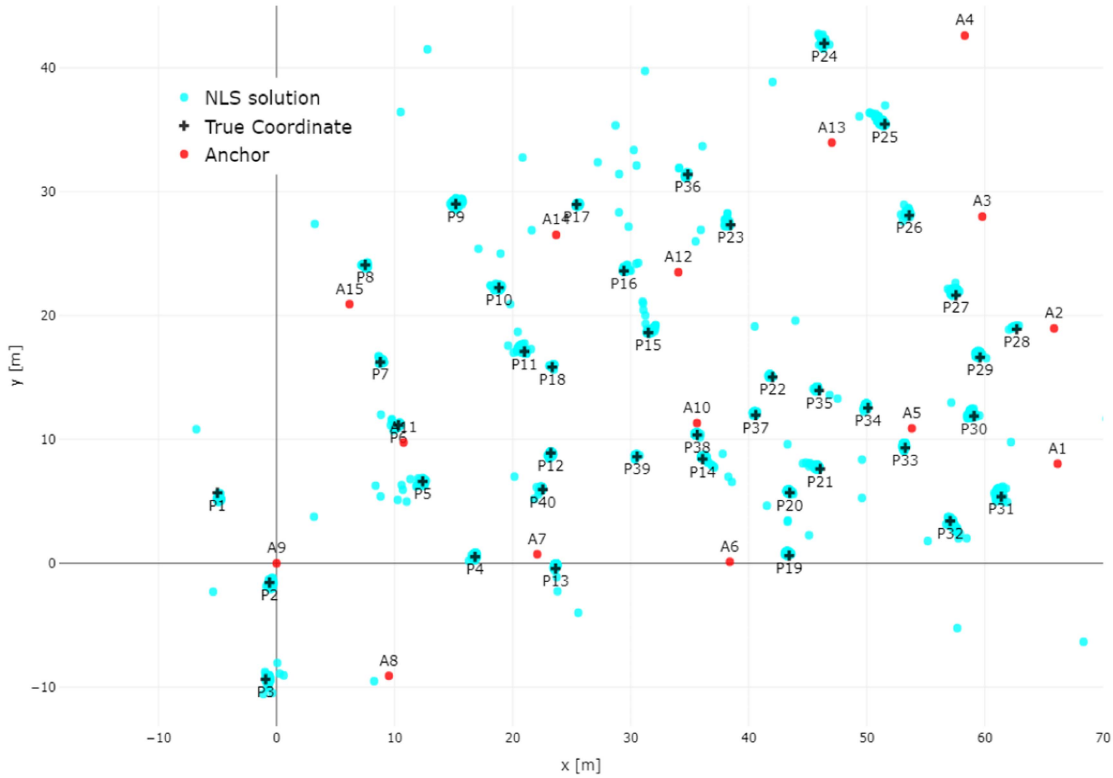


Fig. 17. Overview of measurement campaign in Auroom factory. Measurements were done at 40 separate points around the factory's indoor area.

True coordinates were measured with the Disto S910 measurement device and ranging data were collected using UWB tag with a 10-Hz update rate. The goal was to test the performance of different end coordinate calculation methods, specifically comparing regular filtering methods to those augmented with ML prediction. In addition, no data gathered from the test site was included in ML model training to have unbiased verification of the model.

Considering true coordinates (x_T, y_T, z_T), the following metrics were used to evaluate positioning accuracy and precision: mean location error (MLE), root mean square error (RMSE), distance root mean square error (DRMS), mean radial spherical error (MRSE) and maximum error [38], [39].

1) 2-D metrics

$$\text{MLE}_{2D} = \frac{\sum_{i=1}^{n} \sqrt{(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2}}{n} \tag{34}$$

$$\text{RMSE}_{2D} = \sqrt{\frac{\sum_{i=1}^{n} [(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2]}{n}} \tag{35}$$

$$\text{DRMS} = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{36}$$

$$\text{MAX}_{2D} = \max_{i \in n} (\sqrt{(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2}). \tag{37}$$

2) 3-D metrics

$$\text{MLE}_{3D} = \frac{\sum_{i=1}^{n} \sqrt{(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2 + (z_T - \hat{z}_i)^2}}{n} \tag{38}$$

TABLE I
OVERALL METRICS FOR 2-D POSITIONING

	MLE 2-D [m]	RMSE 2-D [m]	DRMS [m]	Max. error 2-D [m]
NLS	0.46	0.95	0.85	11.16
KF	0.43	0.72	0.57	7.01
AKF + XGB	0.28	0.29	0.11	0.62
AKF + RF	0.28	0.29	0.1	0.55
AKF + RT	0.27	0.28	0.11	0.63
EKF	0.62	0.96	0.78	6.28

$$\text{RMSE}_{3D} = \sqrt{\frac{\sum_{i=1}^{n} [(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2 + (z_T - \hat{z}_i)^2]}{n}} \tag{39}$$

$$\text{MRSE} = \sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}, \tag{40}$$

$$\text{MAX}_{3D} = \max_{i \in n} (\sqrt{(x_T - \hat{x}_i)^2 + (y_T - \hat{y}_i)^2 + (z_T - \hat{z}_i)^2}). \tag{41}$$

Overall statistics summarizing all 40 measurement points can be seen in Tables I and II.

In general, it was challenging to achieve good vertical precision and accuracy in most of the measurement locations. This can be attributed to UWB anchor layout geometry, with anchors located approximately on the same height level, resulting in a poor DOP. Additional difficulties arose from occasional NLOS conditions between anchors and the tag. However, as shown in Table II, the overall vertical position error of approximately 0.5 m is at a similar level as in a previously published work by Laadung et al. [40].

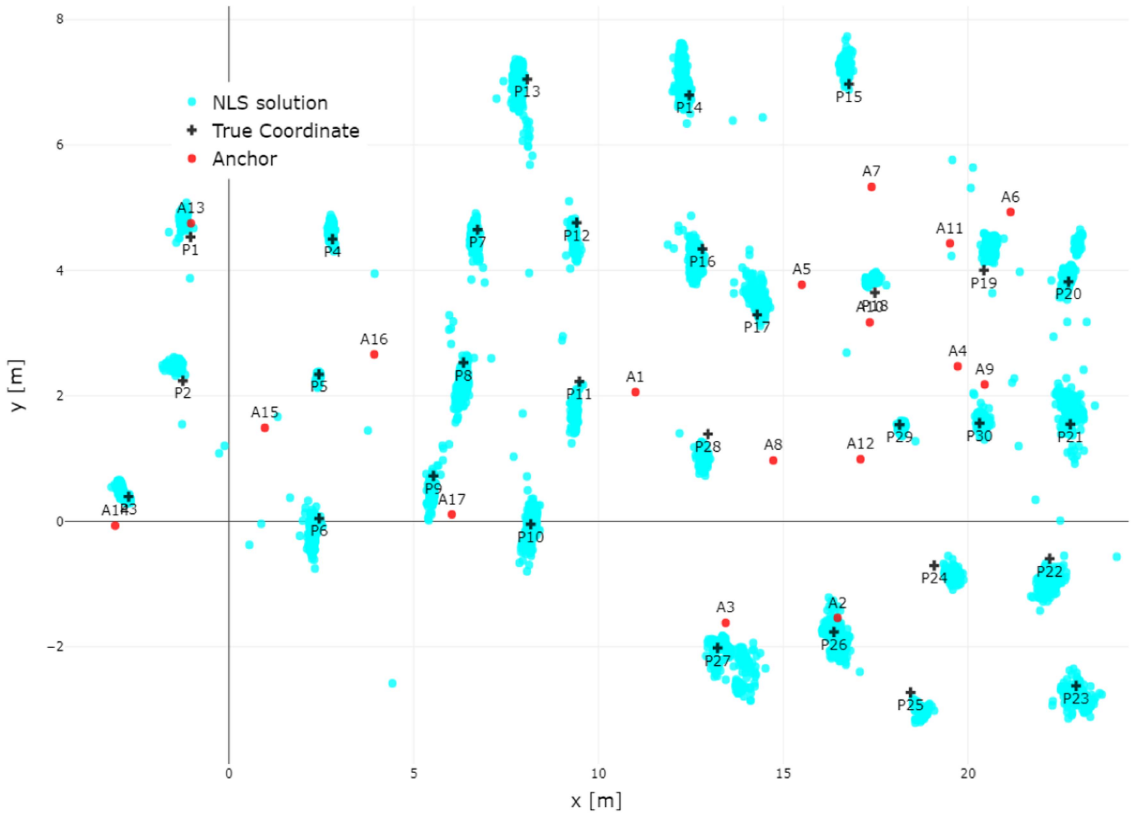


Fig. 18. Overview of measurement campaign in Eliko office. Several measurement points (e.g., 14, 15, 22, and 23) are not surrounded by anchors and have no LOS with them.

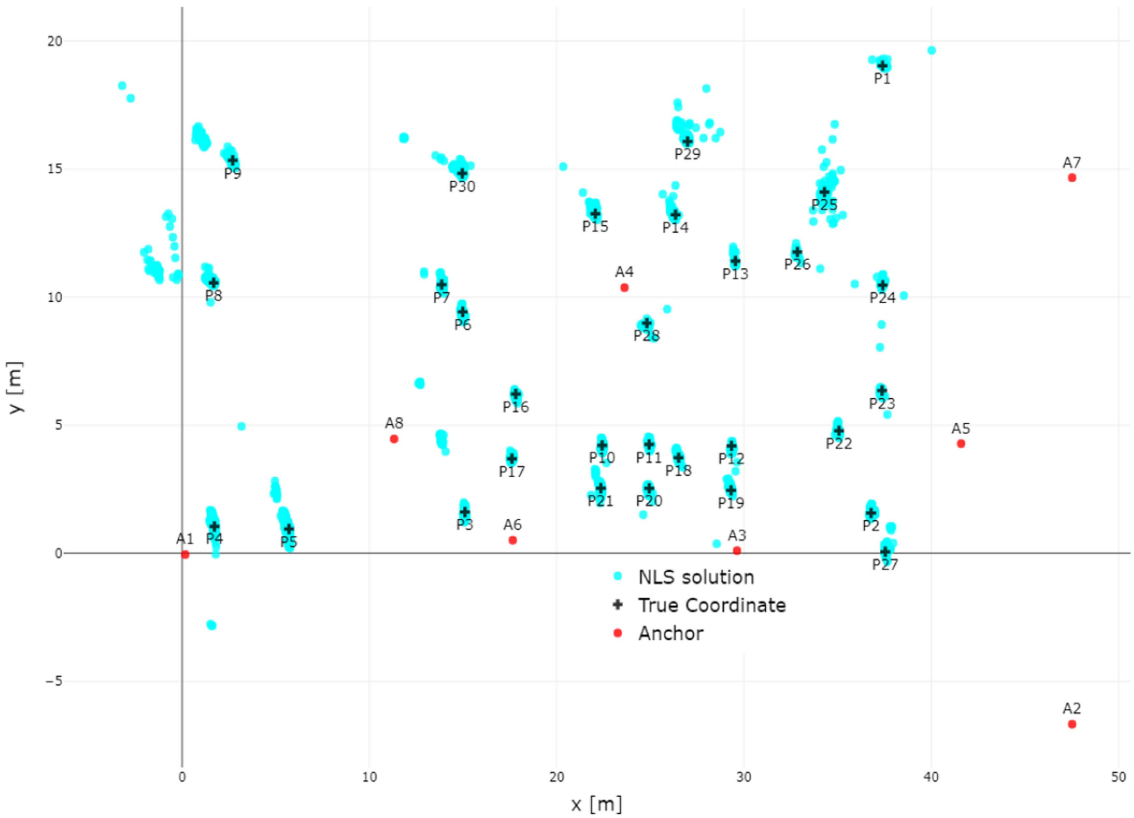


Fig. 19. Overview of measurement campaign in Krah Pipes factory.

TABLE II
OVERALL METRICS FOR 3-D POSITIONING

	MLE 3-D [m]	RMSE 3-D [m]	DRMS [m]	Max. error 3-D [m]
NLS	0.8	1.36	1.17	14.04
KF	0.74	1.05	0.8	8.78
AKF + XGB	0.48	0.5	0.18	0.94
AKF + RF	0.48	0.5	0.18	0.9
AKF + RT	0.51	0.53	0.2	1.07
EKF	2.86	3.26	1.94	11.73

TABLE III
COMPARISON OF END COORDINATE PERFORMANCE METRICS IN 2D POSITIONING FOR PT. 20

	MLE 2D [m]	**RMSE 2D [m]**	**DRMS [m]**	**Max. error 2D [m]**
NLS	0.2	0.24	0.22	1.42
KF	0.16	0.19	0.15	0.85
AKF + XGB	0.14	0.14	0.07	0.24
AKF + RF	0.13	0.14	0.07	0.25
AKF + RT	0.15	0.15	0.06	0.24
EKF	0.25	0.52	0.51	3.22

TABLE IV
COMPARISON OF END COORDINATE PERFORMANCE METRICS IN 3-D POSITIONING FOR PT. 20

	MLE 3-D [m]	RMSE 3-D [m]	DRMS [m]	Max. error 3-D [m]
NLS	0.57	0.65	0.43	2.7
KF	0.53	0.58	0.31	1.9
AKF + XGB	0.41	0.42	0.14	0.67
AKF + RF	0.44	0.45	0.15	0.69
AKF + RT	0.46	0.47	0.12	0.68
EKF	0.65	0.87	0.74	5.74

TABLE V
PERFORMANCE METRICS OF ML MODELS ON THE TEST SET

	RMSE	MSE	MAE
XGB	1.28	1.64	0.36
RF	1.37	1.87	0.46
RT	1.18	1.4	0.33

An example of superimposed end coordinate results can be seen in Figs. 14 and 15 along with respective performance metrics in Tables III and IV. The general location of the point can be seen on overall the map in Fig. 17. It can be seen both visually and statistically that the EKF had the worst performance, especially in 3-D positioning. With many visible outliers, the EKF relies on coordinates calculated straight from noisy ranging data. Furthermore, the EKF does not have any convergence process (i.e., Gauss–Newton iterations), thus relying only on the first calculated end coordinate solution. On the other hand, filtering with ML prediction outperforms non-ML approach in all metrics.

Regarding three different ML algorithms it can be seen that even by applying a simple RT, the overall metrics are better compared to non-ML filtering. ML performance was summarized with the cumulative error distribution in Fig. 16 and metrics for model prediction performance in Table V. The latter includes commonly used regression performance indicators such as RMSE, mean square error (MSE), and mean absolute error (MAE) [41].

TABLE VI
SINGLE ITERATION TIME FOR FILTERING AND PREDICTION

	Min. time [ms]	Mean time [ms]	Max. time [ms]	Ratio
KF	0.04	0.04	0.06	1
AKF + XGB	0.68	0.71	0.72	17.75
AKF + RF	14.77	15.49	16.74	387.25
AKF + RT	0.68	0.72	0.9	18
EKF	0.12	0.14	0.21	3.5

Finally, filtering and ML methods were compared in terms of elapsed time with results shown in Table VI. Benchmarking was done in the R Studio environment using built-in ML libraries $xgboost$, $ranger$, $rpart$, and $microbenchmark$. The hardware specification of the computer was Intel(R) Core(TM) i5-7300 U CPU at 2.60 GHz with 16-GB RAM. It can be seen the amount of delay ML adds to the filtering scheme. An ordinary KF performs the fastest, while the EKF being 3.5 times slower. However, ML prediction adds computational delay, with XGB and RT being approximately 18 times slower than the KF, and RF being the slowest. Finally, the XGB model was also applied in the Eliko RTLS UWB positioning solution using the XGB C Package [42]. The system hardware consisted of Intel(R) Xeon(R) W-2123 CPU at 3.60 GHz with 16-GB RAM. The prediction time delay was approximately 1 ms.

V. CONCLUSION

In this article, it was investigated how different features of ranging residuals and coordinate calculation can be used in UWB-based positioning integrity estimation. These features were described through statistical metrics like those used in literature as well as several novel ones. The goal was to use different ML methods to select features with the biggest informational gain, and based on these selected features, predict the end coordinate offset from the true value. Finally, this error was used as a measure of uncertainty in a coordinate filtering scheme and compared with non-ML-driven filters. It was shown that ML models provide significant improvement in terms of accuracy and precision in both 2-D and 3-D positioning. Overall statistics show that ML-driven filtering has approximately 0.1 m less MLE and 0.3 m smaller DRMS than compared to ordinary KF in 2-D positioning. All of the tested methods were also compared in terms of processing time. ML-driven methods presented a significant delay when compared to ordinary coordinate filtering due to added model-based prediction. However, the processing time was adequate to be used in a high update rate (e.g., 10 Hz) positioning system. In addition, it was seen how the RT algorithm has approximately the same amount of delay as a much more complex XGB, which consists of 150 consecutive boosting trees. Algorithm runtime may be related to the ML library's implementation, hardware specification, and efficiency of the code. Therefore, the actual implementation in a dedicated RTLS system might result in an even smaller processing delay.

ACKNOWLEDGMENT

The authors would like to thank Dr. T. Laadung and A. Fjodorov for their valuable suggestions and feedback.

References

[1] S. Hayward, K. van Lopik, C. Hinde, and A. West, "A survey of indoor location technologies, techniques and applications in industry," *Internet Things*, vol. 20, 2022, Art. no. 100608.

[2] X. Li and Y. Wang, "Research on a factor graph-based robust UWB positioning algorithm in NLOS environments," *Telecommun. Syst.*, vol. 76, pp. 207–217, 2021.

[3] M. A. Al-Ammar et al., "Comparative survey of indoor positioning technologies, techniques, and algorithms," in *Proc. Int. Conf. Cyberworlds*, 2014, pp. 245–252.

[4] M. Ridolfi, A. Kaya, R. Berkvens, M. Weyn, W. Joseph, and E. D. Poorter, "Self-calibration and collaborative localization for UWB positioning systems: A survey and future research directions," *ACM Comput. Surv.*, vol. 54, no. 4, pp. 1–27, 2021.

[5] B. J. Silva and G. P. Hancke, "Non-line-of-sight identification without channel statistics," in *Proc. 46th Annu. Conf. IEEE Ind. Electron. Soc.*, 2020, pp. 4489–4493.

[6] Y. Guo, W. Li, G. Yang, Z. Jiao, and J. Yan, "Combining dilution of precision and Kalman filtering for UWB positioning in a narrow space," *Remote Sens.*, vol. 14, no. 21, 2022, Art. no. 5409.

[7] M. Wang, Z. Chen, Z. Zhou, J. Fu, and H. Qiu, "Analysis of the applicability of dilution of precision in the base station configuration optimization of ultrawideband indoor TDOA positioning system," *IEEE Access*, vol. 8, pp. 225076–225087, 2020.

[8] A. Maali, H. Mimoun, G. Baudoin, and A. Ouldali, "A new low complexity NLOS identification approach based on UWB energy detection," in *Proc. IEEE Radio Wireless Symp.*, 2009, pp. 675–678.

[9] Z. Xiao, H. Wen, A. Markham, N. Trigoni, P. Blunsom, and J. Frolik, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1689–1702, Mar. 2015.

[10] S. Marano, 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.

[11] J. Fan and A. S. Awan, "Non-line-of-sight identification based on unsupervised machine learning in ultra wideband systems," *IEEE Access*, vol. 7, pp. 32464–32471, 2019.

[12] M. J. Bocus and R. J. Piechocki, "Passive unsupervised localization and tracking using a multi-static UWB radar network," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 01–06.

[13] V. Barral, C. J. Escudero, and J. A. García-Naya, "NLOS classification based on RSS and ranging statistics obtained from low-cost UWB devices," in *Proc. 27th Eur. Signal Process. Conf.*, 2019, pp. 1–5.

[14] 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, 2019, Art. no. 5438.

[15] D. Liu, M.-C. Lee, C.-M. Pun, and H. Liu, "Analysis of wireless localization in nonline-of-sight conditions," *IEEE Trans. Veh. Technol.*, vol. 62, no. 4, pp. 1484–1492, May 2013.

[16] Y.-T. Chan, W.-Y. Tsui, H.-C. So, and P.-c. Ching, "Time-of-arrival based localization under NLOS conditions," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 17–24, Jan. 2006.

[17] T. Therneau and B. Atkinson, "RPART: Recursive partitioning and regression trees," R package version 4.1.19, 2022. [Online]. Available: https://CRAN.R-project.org/package=rpart

[18] M. N. Wright and A. Ziegler, "ranger: A fast implementation of random forests for high dimensional data in C and R," *J. Stat. Softw.*, vol. 77, no. 1, pp. 1–17, 2017.

[19] T. Chen et al., "XGBoost: Extreme gradient boosting," R package version 1.6.0.1, 2022. [Online]. Available: https://CRAN.R-project.org/package=xgboost

[20] J. Guillory, D. Truong, and J.-P. Wallerand, "Multilateration with self-calibration: Uncertainty assessment, experimental measurements and Monte-Carlo simulations," *Metrology*, vol. 2, no. 2, pp. 241–262, 2022.

[21] P. Moravek, D. Komosny, M. Simek, and J. Muller, "Multilateration and flip ambiguity mitigation in ad-hoc networks," *Przeglad Elektrotechniczny*, vol. 2012, no. 05b, pp. 222–229, 2012.

[22] K. Liu and Z. Li, "Adaptive Kalman filtering for UWB positioning in following luggage," in *Proc. 34rd Youth Acad. Annu. Conf. Chin. Assoc. Automat.*, 2019, pp. 574–578.

[23] I. Guvenc, S. Gezici, F. Watanabe, and H. Inamura, "Enhancements to linear least squares localization through reference selection and ML estimation," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2008, pp. 284–289.

[24] I. Guvenc, C.-C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2007, pp. 1571–1576.

[25] W. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," Dept. Math. Comput. Sci., Colorado Sch. Mines, Golden, CO, USA, Tech. Rep. MCS-95, vol. 7, 1995, Art. no. 19.

[26] Welcome to Krah Pipes. Accessed: Nov. 15, 2023. [Online]. Available: http://www.krah-pipes.ee/eng/

[27] Decawave Ltd., "DW1000 datasheet." Accessed: Nov. 15, 2023. [Online]. Available: https://www.qorvo.com/products/d/da007946

[28] I. Dotlic, A. Connell, and M. McLaughlin, "Ranging methods utilizing carrier frequency offset estimation," in *Proc. 15th Workshop Positioning, Navigation Commun.*, 2018, pp. 1–6.

[29] Leica Geosystems AG, "Leica DISTO S910." Accessed: Jan. 03, 2024. [Online]. Available: https://shop.leica-geosystems.com/sites/default/files/2019-04/leica_disto_s910_fol_828429_0115_en.pdf

[30] R Core Team, "R: A language and environment for statistical computing," R Foundation for Statistical Computing, Vienna, Austria, 2021. [Online]. Available: https://www.R-project.org/

[31] M. Kuhn, "Building predictive models in R using the caret package," *J. Stat. Softw.*, vol. 28, no. 5, pp. 1–26, 2008. [Online]. Available: https://www.jstatsoft.org/index.php/jss/article/view/v028i05

[32] O. M. Badawy and M. A. B. Hasan, "Decision tree approach to estimate user location in WLAN based on location fingerprinting," in *Proc. Nat. Radio Sci. Conf.*, 2007, pp. 1–10.

[33] C. C. Aggarwal et al. *Data Mining: The Textbook*, vol. 1. Berlin, Germany: Springer, 2015.

[34] Y. Xiong et al., "Seamless global positioning system/inertial navigation system navigation method based on square-root cubature Kalman filter and random forest regression," *Rev. Sci. Instrum.*, vol. 90, no. 1, 2019, Art. no. 015101.

[35] S. Ramraj, N. Uzir, R. Sunil, and S. Banerjee, "Experimenting XGBoost algorithm for prediction and classification of different datasets," *Int. J. Control Theory Appl.*, vol. 9, no. 40, pp. 651–662, 2016.

[36] S. Li and X. Zhang, "Research on orthopedic auxiliary classification and prediction model based on XGBoost algorithm," *Neural Comput. Appl.*, vol. 32, pp. 1971–1979, 2020.

[37] D.-H. Kim, A. Farhad, and J.-Y. Pyun, "UWB positioning system based on LSTM classification with mitigated NLOS effects," *IEEE Internet Things J.*, vol. 10, no. 2, pp. 1822–1835, Jan. 2022.

[38] B. Silva and G. P. Hancke, "Ranging error mitigation for through-the-wall non-line-of-sight conditions," *IEEE Trans. Ind. Inform.*, vol. 16, no. 11, pp. 6903–6911, Nov. 2020.

[39] G. Laveti, G. S. Rao, K. J. Rani, A. Nalinee, and A. M. Babu, "GPS receiver SPS accuracy assessment using LS and LQ estimators for precise navigation," in *Proc. Annu. IEEE India Conf.*, 2014, pp. 1–5.

[40] T. Laadung, S. Ulp, A. Fjodorov, M. M. Alam, and Y. Le Moullec, "Adaptive extended Kalman filter position estimation based on ultra-wideband active-passive ranging protocol," *IEEE Access*, vol. 11, pp. 92575–92588, 2023.

[41] J. S. Kushwah, A. Kumar, S. Patel, R. Soni, A. Gawande, and S. Gupta, "Comparative study of regressor and classifier with decision tree using modern tools," *Mater. Today: Proc.*, vol. 56, pp. 3571–3576, 2022.

[42] XGboost Developers, "C. XGBoost Package." Accessed: Nov. 15, 2023. [Online]. Available: https://xgboost.readthedocs.io/en/stable/c.html

Mihkel Tommingas received professional higher education in 2008 from Tartu Aviation College, Tartu, Estonia, and the M.Sc. degree in telecommunication in 2017 from the Tallinn University of Technology, Tallinn, Estonia, where he is currently working toward the Ph.D. degree in information and communication technology.

He has worked in both the military and civilian sectors. From 2008 to 2015, he worked as a Communications and IT Officer with the Estonian Air Force, and from 2017 to 2021, as a Radio Access Network Measurement Specialist with Elisa Corporation. Since 2021, he has been working as a Researcher with OÜ Eliko Tehnoloogia Arenduskeskus, Tallinn. His current research interests include the improvement of indoor and multinetwork positioning algorithms.

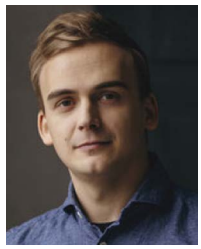
Muhammad Mahtab Alam (Senior Member, IEEE) received the M.Sc. degree in electrical engineering from Aalborg University, Aalborg, Denmark, in 2007, and the Ph.D. degree in signal processing and telecommunication from the INRIA Research Center, University of Rennes 1, Rennes, France, in 2013.

From 2014 to 2016, he was Postdoctoral Researcher with the Qatar Mobility Innovation Center, Qatar. In 2016, he joined as the European Research Area Chair and as an Associate Professor with the Thomas Johann Seebeck Department of Electronics, Tallinn University of Technology, Tallinn, Estonia, where he was elected as a Professor in 2018 and Tenured Full Professor in 2021. Since 2019, he has been the Communication Systems Research Group Leader. He has more than 15 years of combined academic and industrial multinational experiences while working in Denmark, Belgium, France, Qatar, and Estonia. He is an author and coauthor of more than 100 research publications. He is actively supervising a number of Ph.D. and Postdoc Researchers. He is also a contributor in two standardization bodies (ETSI SmartBAN and IEEE-GeeenICT-EECH), including "Rapporteur" of work item: DTR/ SmartBAN 0014. His research interests include the fields of wireless communications-connectivity, mobile positioning, and 5G/6G services and applications.

Dr. Alam has several leading roles as a Principal Investigator in multimillion Euros international projects funded by European Commission (Horizon Europe LATEST-5GS, 5G-TIMBER, H2020 5GROUTES, NATOSPS (G5482), Estonian Research Council (PRG424), and Telia Industrial Grant, etc.

Ivo Müürsepp received the B.Sc. degree in telecommunication cum laude and the M.Sc. and Ph.D. degrees in telecommunication, all from the Tallinn University of Technology (TUT), Tallinn, Estonia, in 2002, 2004, and 2013, respectively.

From 2002 to 2015, he was a Teaching Assistant with the TUT Institute of Radio and Communication Technology, where from 2015 to 2018, he was an Associate Professor. From 2015 to 2020, he was a specialist of IT systems in cyber command with Estonian Defence Forces. Since 2018, he has been a Senior Lecturer with the Thomas Johann Seebeck Institute of Electronics, TUT. His main research interests include mobile communications, mobile positioning, and indoor positioning.



Sander Ulp received the M.Sc. degree in telecommunication and the Ph.D. degree in information and communication technology from the Tallinn University of Technology, Tallinn, Estonia, in 2013 and 2019, respectively.

In 2018, he was a Researcher with OÜ Eliko Tehnoloogia Arenduskeskus, Tallinn, where since 2019, he has been the Chief Technology Officer, which is dedicated to pioneering advancements in indoor positioning research and technologies. His current research interests include distributed estimation, learning and adaptation over networks, digital signal processing, localization technologies, and indoor positioning.