IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 13 September 2024, accepted 28 October 2024, date of publication 4 November 2024, date of current version 19 November 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3490608

RESEARCH ARTICLE

Real-Time Indoor Localization System Based on Wearable Device, Bluetooth Low Energy (BLE) Beacons, and Machine Learning

BAEJAH[1], NUR AHMADI[1,2,3], (Member, IEEE), RAHMAT MULYAWAN[1,3], (Member, IEEE), AND TRIO ADIONO[1,3], (Senior Member, IEEE)

[1]School of Electrical Engineering and Informatics, Bandung Institute of Technology, Bandung 40132, Indonesia
[2]Center for Artificial Intelligence (U-CoE AI-VLB), Bandung Institute of Technology, Bandung 40132, Indonesia
[3]Microelectronics Center, Bandung Institute of Technology, Bandung 40132, Indonesia

Corresponding author: Nur Ahmadi (nahmadi@itb.ac.id)

This work was supported in part by Indonesia Endowment Fund for Education Agency (LPDP) under Grant PRJ-50/LPDP/2020, Grant S-248/LPDP.4/2021, and Grant S-446/LPDP.4/2023.

This work involved human subjects in its research. Approval of all ethical and experimental procedures and protocols was granted by the Research Ethics Committee of the Bandung Institute of Technology, Indonesia under Application No. KEP/II/2022/X/M130723NA/PSPT, and performed in line with the Helsinki Declaration.

ABSTRACT Indoor localization systems are critical in various domains, particularly healthcare, where real-time monitoring of elderly and dementia patients is essential. Current systems face significant challenges in achieving both high accuracy and real-time performance in indoor environments. To address this issue, this study proposes an accurate and real-time indoor localization system that integrates Bluetooth Low Energy (BLE) beacons, wearable device, and advanced machine learning algorithm to enhance room-level localization accuracy. We explored and optimized six machine learning models, including XGBoost, LightGBM, Random Forest, K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Multilayer Perceptron (MLP). A Bayesian optimization framework, Optuna, was used to optimize the hyperparameters of machine learning models. Received Signal Strength Indicator (RSSI) data from 15 participants across 10 rooms were collected and processed for performance evaluation and comparison. Based on the experimental results, XGBoost emerged as the highest performing model, with an average accuracy, precision, recall, and F1-score of 0.91. The complete system demonstrates real-time capability, with an end-to-end execution time of 1,346.27 ms. This highlights the system's potential for practical, accurate, and real-time indoor localization.

INDEX TERMS Indoor localization, BLE beacons, RSSI, wearable device, machine learning.

I. INTRODUCTION

Indoor localization systems are increasingly vital for enhancing home care services, particularly for elderly people and individuals with special needs, such as those suffering from dementia. Dementia is a leading cause of death among the elderly, exacerbated by vulnerabilities to secondary conditions such as infections and falls associated with Alzheimer's disease [1]. In Indonesia, the prevalence of dementia is alarming, with 23.4% of the elderly population in regions like Jakarta and North Sumatra affected [2]. As the elderly population in Indonesia is projected to more than double from 30 million in 2021 to over 60 million by 2045 [3], the urgency for effective monitoring systems is becoming critical.

Despite the pressing need, research on efficient and accurate indoor localization systems for the elderly remains limited. Current systems are often inadequate, with only 25% providing real-time data accessible to caregivers or family members [4]. This deficiency in real-time monitoring

The associate editor coordinating the review of this manuscript and approving it for publication was Eyuphan Bulut.

significantly increases the risk of disorientation and falls among dementia patients, which can lead to fatal outcomes. For instance, between January 2017 and 2019, 293 elderly individuals in Surabaya died as a result of falls, underscoring the critical need for improved monitoring solutions.

In healthcare, these systems are especially vital for monitoring elderly individuals and dementia patients. However, existing systems often fail to deliver the necessary level of accuracy and real-time performance. Bluetooth Low Energy (BLE) has emerged as a promising technology due to its low cost, energy efficiency, and capability to provide room-level accuracy. The potential of BLE is further enhanced when combined with machine learning algorithms, which can improve the system's ability to handle signal interference and dynamic environmental conditions.

Previous research has explored various approaches to indoor localization. Meng and Li [5] investigated cost-effective wearable localization by integrating Ultra-Wideband (UWB) and Inertial Measurement Unit (IMU) technologies, demonstrating that sensor fusion can enhance accuracy. Zhang and Sun [6] provided a comprehensive review of data-driven models for indoor localization, with a focus on the challenges and future directions, particularly in elderly care applications. Zafari et al. [7] conducted a comprehensive survey of indoor localization systems and technologies, including BLE and Wi-Fi-based methods, while Nguyen et al. [8] reviewed wireless positioning techniques, emphasizing the accuracy and limitations of technologies such as RFID, UWB, and BLE. Xu et al. [9] improved the K-Nearest Neighbor (KNN) algorithm for indoor positioning, resulting in enhanced speed and accuracy, while Gao et al. [10] proposed the use of Multilayer Perceptron (MLP), which has proven effective in handling the non-linear data patterns commonly found in indoor environments.

While RFID and UWB have been explored extensively, their limitations in terms of cost and deployment complexity make them less suitable for widespread real-time monitoring solutions. BLE, in combination with advanced machine learning models, offers a more scalable and accurate alternative [11], [12]. Advanced models such as XGBoost and LightGBM are known for their speed and accuracy, particularly in handling large-scale data, while MLP is effective in dealing with non-linear data patterns, which are common in indoor environments.

This study aims to bridge this gap by proposing a real-time indoor localization system that integrates BLE beacons, wearable devices, and machine learning algorithms. Received Signal Strength Indicator (RSSI) data from the BLE beacons will be received by the wearable device and used as the input for machine learning models. Multiple machine learning models, including XGBoost, LightGBM, Random Forest, KNN, SVM, and MLP, are optimized and evaluated to find the best performing model in terms of accuracy and inference time. The contributions of this study are threefold: the creation of dataset which is made publicly available, a comprehensive exploration and optimization of various machine learning algorithms, and the end-to-end, real-time implementation of indoor localization system based on the highest performing machine learning model.

The remainder of this paper is organized as follows. Section II describes the methodology which includes data acquisition, data processing, machine learning model exploration, training, and evaluation. Section III presents details on the experimental results and discussion. The conclusion is drawn in Section IV.

II. METHODS

A. DATA ACQUISITION

The data collection procedure is shown in Fig. 1. The layout in Fig. 2a represents the experimental setup conducted in a campus building. Fifteen university students were recruited as participants for this experiment. The data, in the form of RSSI, were collected using 10 BLE beacons (iBKS 105) [13], which were placed at 10 distinct points throughout the experimental layout. Each point represented a different room in the building, as shown in Fig. 2a. For instance, the rooms labeled A, B, C, etc., correspond to different spaces, with each beacon strategically positioned to capture RSSI data for that specific area. RSSI measures the power level received by the BLE antenna on the ESP32-based wearable device, which functioned as the Base Station Node (BSN). The wearable

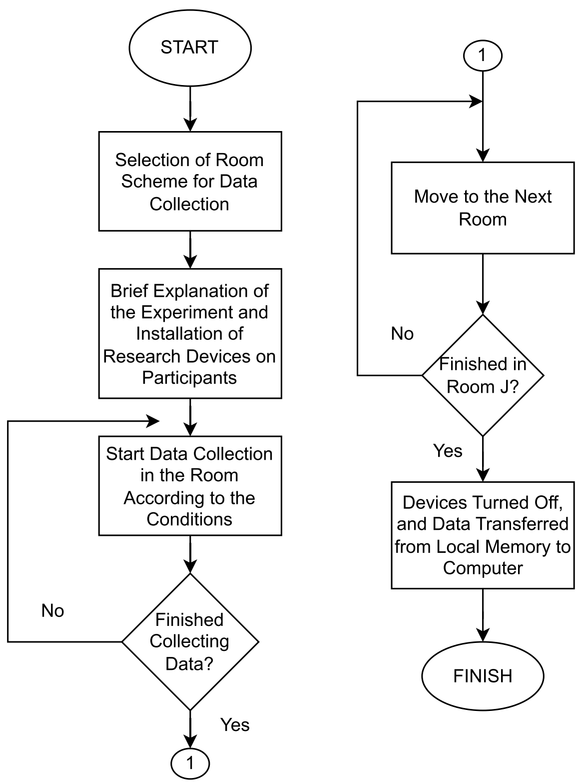

FIGURE 1. Data collection procedure.

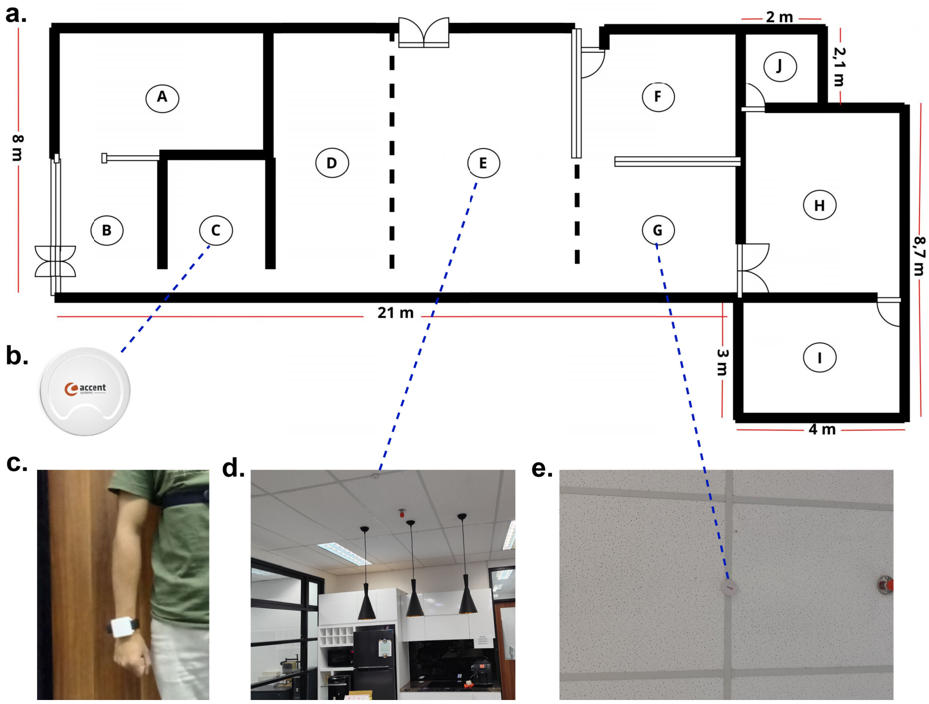

FIGURE 2. (a) Layout of the data collection building. (b) iBKS 105 BLE beacon. (c) Participant wearing the ESP32-based wearable device. (d) Inside view of a data collection room. (e) iBKS 105 BLE beacon mounted on the ceiling.

device, worn by participants (as shown in Fig. 2c), stored the RSSI data transmitted from the BLE beacons. The Base Station Node (BSN) sent a feed packet to the reference node, which was then forwarded to the target node. The target node identified the BLE Beacon ID, determined its origin node, and then transmitted the RSSI data back to the BSN, where it was stored in the internal memory of the ESP32 [14]. Fig. 2b shows the iBKS 105 BLE beacon used to capture and transmit RSSI signals. The beacons were installed in the rooms' ceiling where they could not be easily reached by participants (Fig. 2e), preventing any interference or disturbance. Fig. 2d provides an internal view of one of the rooms where data were collected, from the perspective of a participant.

B. DATA PREPROCESSING

Additional validation experiments were conducted to determine the threshold value, which was subsequently applied to clean the data before preprocessing. In each room, Received Signal Strength Indicator (RSSI) data were collected at 10–15 critical points representing the signal strength boundaries of the Bluetooth signals received by the scanning device. The choice of 10 critical points as the lower limit was determined to ensure sufficient coverage of the entire experimental room. The variation in the number of critical points (10–15) was based on the size of each room, with larger rooms requiring more points for comprehensive coverage. The collected data were labeled as BLE01–BLE10, corresponding to the respective room, with BLE01 representing Room 01. Subsequently, rows with more than eight NaN values (indicating signal values below −100) were removed, as they indicated that the majority of BLE beacons were not detected. The 'room' column names were converted from string labels to numeric values using the Label Encoder from the Scikit-learn library for further analysis. The cleaned data were then divided into three sets: 80% for training, 10% for validation, and 10% for testing. The validation set was used during the optimization process to fine-tune model parameters and identify the most suitable model configuration, ensuring generalization and minimizing the risk of overfitting. The final test set was used to evaluate the model's performance, providing an unbiased assessment of accuracy on data not previously seen by the model.

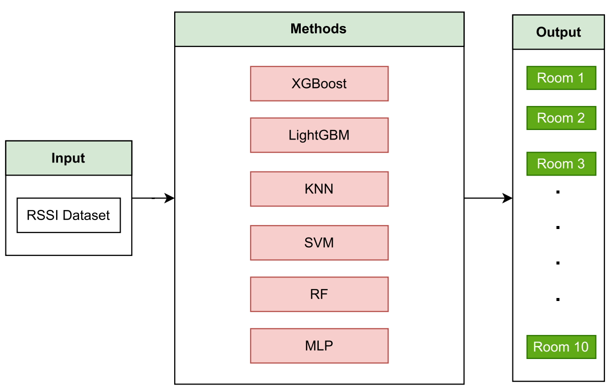

FIGURE 3. Illustration of different machine learning models.

C. MODEL EXPLORATION

In this study, various machine learning models were explored and optimized, as illustrated in Fig. 3. The models include XGBoost [15], LightGBM [16], Random Forest [17], K-Nearest Neighbor (KNN) [9], Support Vector Machine (SVM) [18], and Multilayer Perceptron (MLP) [10]. A description of each model and its optimization process is provided below.

1) XGBoost

XGBoost (Extreme Gradient Boosting), initially developed by Chen and Guestrin [15], is an ensemble method based on decision trees. The performance of the model is improved through gradient boosting, where trees are built sequentially, with each subsequent tree correcting the errors of the previous one. A significant advantage of XGBoost is its ability to handle overfitting effectively through L1 (Lasso) and L2 (Ridge) regularization, which makes it more resilient to noisy data. Each tree is grown level-wise, ensuring balanced and uniform growth across all nodes. The final prediction is computed by summing the outputs of all trees.

TABLE 1. Hyperparameters for XGBoost and LightGBM.

Hyperparameter	Description
n_estimators	Number of gradient boosted trees
max_depth	Maximum tree depth for base learners
learning_rate	Boosting learning rate
subsample	Subsample ratio of the training instance
reg_alpha	L1 regularization term on weights
reg_lambda	L2 regularization term on weights

In this study, XGBoost (v2.1.1) was optimized using hyperparameters such as the number of trees, maximum depth, learning rate, subsample ratio, and regularization values (see Table 1). XGBoost was selected for its efficiency in handling complex, non-linear data, and scalability to large datasets [6]. The equation for XGBoost optimization is shown in Eq. 1.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F} \tag{1}$$

where K represents the total number of trees, and $f_k(x_i)$ denotes the prediction from the k-th tree on instance x_i.

2) LightGBM

LightGBM (Light Gradient-Boosting Machine), developed by Guolin Ke et al. [16], is designed to provide faster training times and lower memory consumption. LightGBM uses a histogram-based decision tree algorithm, which reduces computational costs by discretizing continuous features into bins, leading to more efficient training and lower memory usage. Unlike XGBoost, LightGBM employs a leaf-wise growth strategy, wherein nodes are added to the leaf with the largest error, resulting in deeper, more specialized trees. The final prediction is obtained by aggregating the outputs of all trees.

LightGBM was chosen for its capability to handle large datasets with higher efficiency compared to other models [19]. The hyperparameters were tuned to optimize the number of trees and maximum depth (see Table 1). The decision function for LightGBM is expressed in Eq. 2.

$$y_i = \sum_{j=1}^{J} w_j h_j(x_i) \tag{2}$$

where J is the number of leaves, and $h_j(x_i)$ represents the decision function of the j-th leaf.

3) RANDOM FOREST

Random Forest [17] is an ensemble learning method that builds multiple decision trees during training and aggregates the outputs through majority voting (for classification tasks). The hyperparameters were tuned to optimize the number of trees and maximum depth, and the model was implemented using Scikit-learn (v1.5.1).

Random Forest was selected due to its robustness in reducing overfitting by combining the outputs of multiple decision trees [20]. The general equation for Random Forest prediction is presented in Eq. 3.

$$\hat{y}_i = \frac{1}{T} \sum_{t=1}^{T} h_t(x_i) \tag{3}$$

where T is the total number of trees, and $h_t(x_i)$ represents the prediction from the t-th tree for instance x_i.

4) K-NEAREST NEIGHBOR (KNN)

K-Nearest Neighbor (KNN) is a non-parametric algorithm that classifies data points based on the K-nearest neighbors in the feature space. Various values of K were evaluated to identify the optimal configuration. The KNN model was implemented using Scikit-learn (v1.5.1).

KNN was selected for its simplicity and its ability to effectively classify data based on local distributions [11]. The equation for KNN prediction is provided in Eq. 4.

$$\hat{y}_i = \frac{1}{K} \sum_{k=1}^{K} y_k \tag{4}$$

where y_k represents the label of the k-th nearest neighbor of instance x_i.

5) SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine (SVM) [18] is a supervised learning model that identifies the optimal hyperplane to separate classes in a high-dimensional space. Both linear and kernel-based SVMs (using RBF and polynomial kernels) were evaluated. The regularization parameter C was adjusted to balance the trade-off between maximizing the margin and minimizing classification errors. The model was implemented using Scikit-learn (v1.5.1).

SVM was chosen due to its strong performance in high-dimensional datasets and its ability to generate clear decision boundaries [21]. The optimization objective for SVM is given by Eq. 5.

$$\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(w^T x_i + b)) \tag{5}$$

where w is the weight vector, b is the bias, and y_i is the label for instance x_i.

6) MULTILAYER PERCEPTRON (MLP)

Multilayer Perceptron (MLP) is an artificial neural network comprising multiple layers of nodes, where each node is fully connected to the nodes in the subsequent layer. A feedforward MLP model with one input layer, several hidden layers, and one output layer was utilized. Hyperparameters such as the number of hidden layers, neurons per layer, learning rate, and activation function were tuned to optimize performance [22].

MLP was chosen for its ability to model complex, non-linear relationships that may not be easily captured by other models [23]. The general equation for MLP is given in Eq. 6.

$$\hat{y}_i = f(W_2 f(W_1 x_i + b_1) + b_2) \tag{6}$$

where f represents the activation function, W_1 and W_2 are weight matrices, and b_1 and b_2 are bias vectors.

D. HYPERPARAMETER OPTIMIZATION

To optimize the machine learning models for indoor localization, we employed the Optuna library [24], which utilizes a Bayesian optimization approach to efficiently explore the hyperparameter space. The key hyperparameters optimized for each model, as described in the previous subsection, included:

- **n_estimators**: Number of trees (for XGBoost, LightGBM, Random Forest).
- **max_depth**: Maximum depth of trees (for XGBoost, LightGBM, Random Forest).
- **learning_rate**: Learning rate for gradient-boosting models (for XGBoost, LightGBM).
- **subsample**: Subsampling ratio (for XGBoost, LightGBM).
- **C**: Regularization parameter (for SVM).
- **n_neighbors**: Number of nearest neighbors (for KNN).
- **n_layers, n_neurons**: Number of layers and neurons per layer (for MLP).
- **activation_function**: Activation function for MLP (ReLU).

The hyperparameter optimization process using Optuna is illustrated in Fig. 4. Optuna's Bayesian optimization method searches the hyperparameter space in a guided manner, learning from previous trials to focus on the most promising regions. By leveraging this approach, we were able to find optimal hyperparameter combinations more efficiently than through grid or random search methods. Additionally, the model was refined using Focal Loss, a loss function specifically designed to address class imbalance issues often encountered in indoor localization datasets. This adjustment ensured that the model performed well across both majority and minority classes, which is critical in real-time localization settings where certain locations might be underrepresented. After selecting the best hyperparameters, the machine learning models were trained and validated using 10-fold cross-validation. This process involved splitting the dataset into 10 parts, training the model on 9 parts, and using the remaining part for validation. This process was repeated 10 times, and the average score from these iterations was used to assess the model's generalization ability and prevent overfitting.

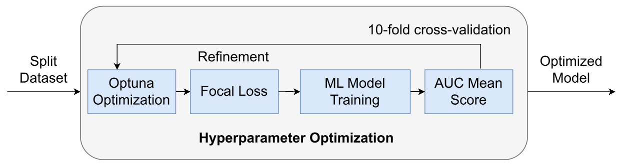

FIGURE 4. Hyperparameter optimization using Optuna.

E. EVALUATION METRICS

Several performance metrics are used including accuracy, precision, recall, and F1-score which can be derived from the correct and incorrect classification results: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). Accuracy is defined as the proportion of correct predictions (both positive and negative) out of the total predictions. Precision represents the proportion of true positive predictions out of all positive predictions made. Recall indicates the proportion of true positive cases detected by the model, while the F1-score represents the harmonic mean of precision and recall. The formula to calculate the above metrics are given below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{9}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

F. REAL-TIME SYSTEM IMPLEMENTATION

Fig. 5 depicts the schematic diagram of the complete real-time system implementation. The system utilizes BLE beacons (IBKs 105) for continuously transmitting RSSI data, which is subsequently received, preprocessed, and cleaned by an ESP32-based wearable device. The cleaned data is then transmitted via Message Queuing Telemetry Transport (MQTT) protocol to a Raspberry Pi 4 Model B [25] for room location prediction (inference). The best performing machine learning model during model exploration and optimization is then selected and deployed to the Raspberry Pi. The total time needed to perform end-to-end model inference is then calculated and reported.

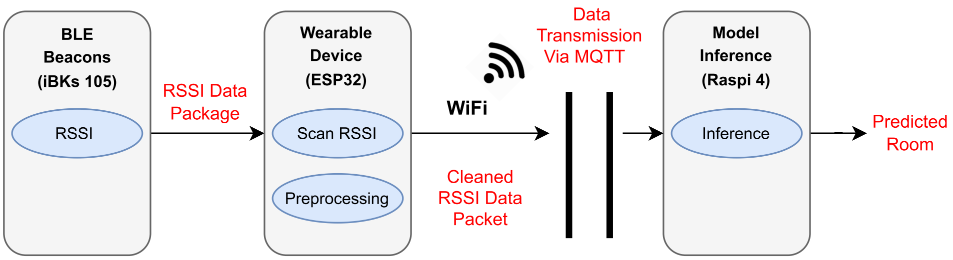

FIGURE 5. Schematic diagram of real-time implementation.

III. RESULTS AND DISCUSSION

A. CREATED DATASET

The collected data was cleaned and processed based on the threshold value of RSSI derived from additional validation experiments. After data cleaning, the number of samples decreases from 21,975 to 12,358. This dataset encompasses 10 classes (rooms) with data formatted to include timestamp, RSSI value, and room name, as specified in Table 2.

TABLE 2. Dataset description.

Specification	Value
Number of classes (rooms)	10
Number of BLE beacons	10
Data attributes	Timestamp, BLE ID, RSSI, Room
Number of raw samples (rows)	21,975
Number of cleaned samples (rows)	12,358

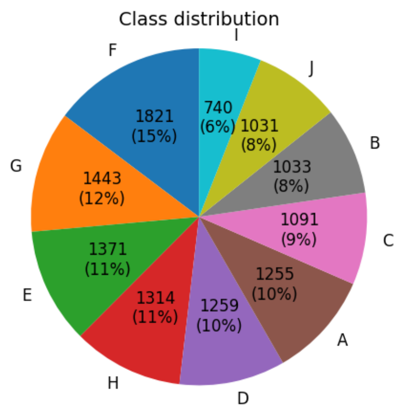

FIGURE 6. Class distribution of the dataset.

The class distribution of the dataset is illustrated in Fig. 6. It shows that the class proportions are relatively balanced, with most classes having proportions between 8% and 12%. However, imbalances are observed in class F (15%) and class I (6%). This is partly due to the variations in room size. Such imbalances might reduce model accuracy for class I and introduce bias towards class F. These imbalances can be mitigated through techniques such as oversampling or undersampling, the selection of appropriate algorithms that are tolerant to imbalances, etc. [4]. The dataset can be accessed from a public repository at https://zenodo.org/records/13317046 [26].

B. PERFORMANCE COMPARISON

We employed hyperparameter optimization using Optuna library to identify optimal hyperparameter combinations for machine learning models. The resulting optimized hyperparameters for XGboost, LightGBM, and Random Forest models are shown in Table 3. For KNN and SVM models, the optimized hyperparameters were N=20 and C=0.361, respectively. For MLP models, the optimized hyperparameters were number of layers=3, number of units (neurons) at first layer=80, number of units at the second layer=109, number of units at the third layer=95, activation function=ReLU, and learning rate=0.0011.

TABLE 3. Optimized hyperparameters for XGBoost, LightGBM and RF models.

Hyperparameter	XGBoost	LightGBM	RF
n_estimators	468	382	93
max_depth	7	3	25
learning_rate	0.160	0.258	-
subsample	0.849	0.880	-
reg_alpha	3.270	1.439	-
reg_lambda	1.267	1.512	-

TABLE 4. Performance comparison across ML models.

Method	Precision	Recall	F1-Score	Accuracy
XGBoost	**0.91**	**0.91**	**0.91**	**0.91**
LightGBM	0.91	0.91	0.91	0.90
RF	0.69	0.69	0.69	0.69
KNN	0.37	0.38	0.37	0.37
SVM	0.41	0.40	0.38	0.39
MLP	0.60	0.61	0.60	0.61

Using the resulting optimized hyperparameters, we compared the performance of six models. The performance comparison is summarized in Table 4. We found that XGBoost performs slightly better than LightGBM, and significantly better than the remaining models. XGBoost yielded an average performance of 0.91 across 4 metrics (accuracy, precision, recall, and F1-score). XGBoost and LightGBM are specifically designed to handle complex and high-dimensional data more effectively through the boosting process [5]. These models build iteratively, correcting the errors of previous models, which allows them to capture non-linear relationships and feature interactions more effectively. Features such as automatic feature scaling and strong regularization capabilities also contribute to their superior performance compared to other algorithms. The lower performance of other methods, despite optimization with Optuna, may be attributed to inherent algorithmic limitations and the constraints of available parameters for optimization.

TABLE 5. **Performance evaluation of the XGboost model.**

Class	Precision	Recall	F1-Score	Accuracy
A	0.92	0.94	0.93	0.94
B	0.93	0.88	0.91	0.92
C	0.94	0.94	0.94	0.94
D	0.85	0.96	0.90	0.91
E	0.97	0.85	0.91	0.92
F	0.91	0.96	0.93	0.94
G	0.92	0.86	0.89	0.91
H	0.87	0.89	0.88	0.90
I	0.88	0.80	0.84	0.86
J	0.92	0.97	0.94	0.96
Macro avg	0.91	0.91	0.91	0.91

KNN and SVM have inherent limitations in handling complex and non-linear data commonly encountered in indoor localization problems [6]. For instance, KNN relies heavily on distance metrics and can be ineffective in high-dimensional feature spaces. SVM may struggle to find the optimal hyperplane if the data are highly non-linear. Each algorithm requires specific parameters to be well-tuned to achieve optimal performance, such as the number of neighbors in KNN or the regularization parameter in SVM. While Optuna can assist in finding optimal parameters, there are limits to how much these parameters can enhance performance [27]. MLP had relatively low performance across four metrics. This could be due to that MLP model need larger, cleaner, and more consistent data for optimal performance [7]. The dataset used in this study is not large and tends to exhibit high variability which is influenced by physical obstructions, interference, and distance. Expanding the hyperparameter searching space and increasing the number of iterations during hyperparameter optimization in Optuna may provide more opportunities to find optimal combinations which can improve the model's performance [28].

To understand how XGBoost model performs for each class, we can see the detail of multi-class classification results in Table 5. The confusion matrix for the XGBoost model in Fig. 7 demonstrates strong generalization capabilities with high precision and recall across multiple classes. However, the model has the difficulty to differentiate classes D and E due to similarities in RSSI patterns between both classes within contiguous spaces. The higher recall value (0.96) and lower precision value (0.85) suggest that the model effectively classifies class D but sometimes misclassifies other classes as D. Class I exhibits the lowest precision, recall, and F1-score might be due to limited data, highlighting the effect of class imbalance. The future work may address this issue through several techniques such as data augmentation to introduce environmental variability during training, oversampling or undersampling to make sure the data is balanced.

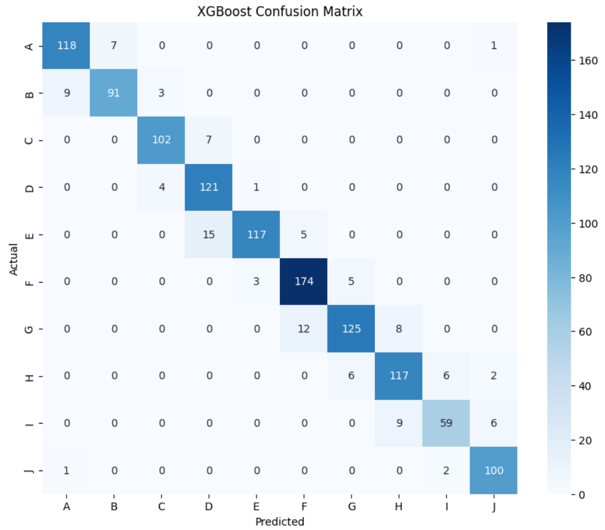

FIGURE 7. **Confusion matrix of the XGboost model.**

TABLE 6. **XGBoost real-time implementation time.**

Process	Execution Time (ms)
BLE packet scanning process	116.00
BLE data preprocessing and cleaning	138.38
Data transmission from ESP32 to Raspberry	224.00
Model loading time	38.85
Model inference time	815.24
Inference result visualization time	13.80
Total Execution Time	1,346.27

C. REAL-TIME INFERENCE

The implementation of a real-time system involves a series of components and stages required to detect and process data from BLE beacons in each room. The breakdown of the overall execution time for end-to-end real-time indoor localization system is detailed in Table 6. The overall execution time takes around 1.35 seconds. The process that takes the longest execution time is XGBoost model inference. XGBoost is chosen as the model to be deployed into real-time system it yields the highest accuracy results. It takes 815.24 ms to perform inference. The second longest execution time is for data transmission from ESP32 to Raspberry Pi which takes 224.00 ms. The data transmission is influenced by network quality, distance between devices, and signal interference [29]. It is important to note that the execution time depends on the devices' hardware capability being used and the efficiency of the model [30]. The ML model is selected based on the real-time implementation requirements that emphasize accuracy and real-time execution. The specific requirements of the real-time application align with the response time definition of 1-1.5 seconds [31] for room localization systems. Our proposed system can meet this real-time requirement while providing high accuracy.

Future works can be directed towards optimizing the real-time model implementation by reducing execution time in each processing stage to achieve a faster real-time response. Considering the use of more efficient models can increase speed without significantly sacrificing accuracy. Additionally, improving data preprocessing and communication pathways between wearable device (e.g. ESP32) and (ML inference device (e.g. Raspberry Pi) can help reduce the overall execution time.

IV. CONCLUSION

In this study, we developed a real-time indoor localization system that leverages BLE beacons, wearable devices, and machine learning algorithms to accurately determine the location of individuals within an indoor environment. We collected, cleaned, and processed RSSI data from 15 participants across 10 different rooms. Through extensive evaluation, we explored, optimized, and compared the performance of six machine learning algorithms. Among these, the XGBoost algorithm demonstrated superior performance, achieving an average accuracy, precision, recall, and F1-score of 0.91. The XGBoost model's inference time was 815.24 ms, while the end-to-end system implementation, incorporating both the ESP32 and Raspberry Pi, completed the localization task in 1,346.27 ms. These results confirm that our proposed system is not only capable of operating in real-time but also delivers high accuracy, making it a viable solution for practical indoor localization applications.

REFERENCES

[1] A. D. Int. (May 12, 2019). *World Alzheimer Report 2019: Attitudes To Dementia*. [Online]. Available: https://www.alzint.org

[2] *Survey Hasil Demensia Di Jakarta Dan Sumatera*, STRiDE, Indonesia, 2023.

[3] Badan Pusat Statistik. *Proyeksi Penduduk Indonesia 2020-2050*. Accessed: May 12, 2024. [Online]. Available: https://www.bps.go.id

[4] M. Johnson, "Real-time monitoring systems for the elderly: A review of current technologies," *J. Geriatric Care*, vol. 45, no. 2, pp. 123–136, 2020.

[5] H. Zhang, Z. Zhang, N. Gao, Y. Xiao, Z. Meng, and Z. Li, "Cost-effective wearable indoor localization and motion analysis via the integration of UWB and IMU," *Sensors*, vol. 20, no. 2, p. 344, Jan. 2020.

[6] Y. Zhang and W. Sun, "Understanding data-driven models for indoor localization: Applications, challenges, and future directions," *IEEE Access*, vol. 8, pp. 184616–184633, 2020.

[7] X. Gu, F. Ren, and J. Zhang, "A survey on deep learning for indoor localization," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1104–1135, Feb. 2021.

[8] T.-H. Nguyen, D. Tien, and P. Pham, "Wireless indoor positioning based on machine learning: A survey," *IEEE Access*, vol. 10, pp. 17525–17541, 2022.

[9] J. Xu, J. Gao, S. Wu, and S. He, "Improved k-nearest neighbors algorithm based on kd tree," *IEEE Access*, vol. 8, pp. 21582–21588, 2020.

[10] Y. Gao, X. Zhu, M. Zhang, and Y. Zhang, "Indoor localization using multilayer perceptron," *IEEE Access*, vol. 8, pp. 211951–211961, 2020.

[11] A. A. Mohamed, M. N. Marson, and H. K. Taha. (2024). *Indoor Positioning in Smart Healthcare Environments Using BLE Technology: Current Trends and Future Prospects*. Accessed: Aug. 11, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9263591

[12] M. D. Kumar, L. V. Ramos, and J. I. Hernandez. *Wearable and IoT-Based Solutions for Elderly Care Monitoring: A Comprehensive Review*. Accessed: Aug. 11, 2024. [Online]. Available: https://www.mdpi.com/1424-8220/22/4/998

[13] Accent Systems. (2024). *IBKS 105*. Accessed: Aug. 11, 2024. [Online]. Available: https://accent-systems.com/product/ibks-105/

[14] Espressif Syst. (2024). *ESP32 DevKits*. Accessed: Aug. 11, 2024. [Online]. Available: https://www.espressif.com/en/products/devkits

[15] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[16] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 3146–3154.

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

[18] X. Zhu, Z. Liu, and M. Zhang, "Advances in support vector machines for large-scale classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4535–4548, Oct. 2021.

[19] R. Montoliu, E. Sansano, A. Gascó, O. Belmonte, and A. Caballer, "Indoor positioning for monitoring older adults at home: Wi-Fi and BLE technologies in real scenarios," *Electronics*, vol. 9, no. 5, p. 728, Apr. 2020.

[20] M. Ren and F. Zhao, "Random forest-based methods for robust overfitting reduction," *J. Data Mining Knowl. Discovery*, vol. 58, no. 4, pp. 789–802, Apr. 2021.

[21] X. Li and W. Zhang, "Improved SVM for high-dimensional data and clear decision boundaries," *Pattern Recognit. Lett.*, vol. 45, no. 6, pp. 123–131, 2022.

[22] H. Yang, D. Kim, and J. Lee, "Optimization techniques for deep learning models using the optuna library," *J. Comput. Sci.*, vol. 54, pp. 101–115, Aug. 2021.

[23] Y. Gao and L. Zhang, "Deep learning based on mlp for complex, non-linear data modeling," *Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 215–225, 2021.

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 2623–2631.

[25] Raspberry Pi. (2024). *Raspberry Pi 4 Model B*. Accessed: Aug. 11, 2024. [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[26] Baejah, N. Ahmadi, R. Mulyawan, and T. Adiono, "Dataset for real-time indoor localization system based on wearable device, bluetooth low energy (BLE) beacons, and machine learning [data set]," Zenodo, 2024, doi: 10.5281/zenodo.13317046.

[27] Z. Zhang and Q. Ma, "A survey of data-driven indoor localization in IoT," *IEEE Internet Things J.*, vol. 8, no. 11, pp. 9146–9163, Nov. 2021.

[28] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 4765–4774.

[29] A. Gerstlauer, J. Peng, and M. Schleifer, "Latency and performance optimization for networked embedded systems," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 321–334, Feb. 2013.

[30] A. Burns and A. Wellings, *Real-Time Systems and Programming Languages: Ada 95, Real-Time Java, and Real-Time POSIX*, 3rd ed., Reading, MA, USA: Addison-Wesley, 2021.

[31] A. L. Dias, A. C. Turcato, G. S. Sestito, D. Brandao, and R. Nicoletti, "A cloud-based condition monitoring system for fault detection in rotating machines using profinet process data," *Comput. Ind.*, vol. 126, 2021, Art. no. 103394.

BAEJAH received the bachelor's degree in electrical engineering from the Institut Teknologi Bandung (ITB), Indonesia, in 2023, where she is currently pursuing the master's degree in electrical engineering, focusing on IoT applications for industrial and societal challenges. Her research interest includes leveraging IoT for industrial applications, particularly in predictive maintenance systems for machinery, inspired by her undergraduate research on predictive maintenance systems for mixing machines in FMCG companies. Her goal is to develop innovative IoT solutions that enhance operational efficiency and address societal needs.

NUR AHMADI (Member, IEEE) received the B.Eng. degree in electrical engineering from Bandung Institute of Technology (ITB), Indonesia, in 2011, the M.Eng. degree in communication and integrated systems from Tokyo Institute of Technology, Japan, in 2013, and the Ph.D. degree in electrical and electronic engineering from the Imperial College London, U.K., in 2020. His Ph.D. research focused on signal processing and deep learning for intracortical brain–machine interfaces. He is currently with the Center for Artificial Intelligence and School of Electrical Engineering and Informatics, ITB. His current research interests include biomedical signal processing, artificial intelligence/machine learning, brain–machine interfaces, neurotechnology, and digital and embedded systems. He received Travel Grant Award at IEEE UKCAS 2018 and Young Researchers Poster Award at IEEE BioCAS 2019.

RAHMAT MULYAWAN (Member, IEEE) received the B.Eng. degree in EE from ITB, Indonesia, in 2008, and the M.Sc. degree in EE from TU Delft, The Netherlands, in 2011. His research interests include intelligent signal processing, MIMO systems, and transceiver design for optical wireless communications. He is a member of the Microelectronics Centre, ITB.

TRIO ADIONO (Senior Member, IEEE) received the B.Eng. degree in electrical engineering and the M.Eng. degree in microelectronics from the Institut Teknologi Bandung, Indonesia, in 1994 and 1996, respectively, and the Ph.D. degree in VLSI design from Tokyo Institute of Technology, Japan, in 2002. He is currently a Professor with the School of Electrical Engineering and Informatics and also the Head of the IC Design Laboratory, Microelectronics Center, Institut Teknologi Bandung. He holds a Japanese Patent on a high quality video compression systems. His research interests include VLSI design, signal and image processing, VLC, smart cards, and electronics solution design and integration.